



TESIS - TE - 142599

OPTIMASI UNTUK PERILAKU PAHLAWAN BERBASIS NSGA - II

CINDY TAURUSTA
NRP. 2213205023

DOSEN PEMBIMBING

Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.
Dr. Eko Mulyanto Yuniarno, S.T., M.T.

PROGRAM MAGISTER

BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA

KONSENTRASI TEKNOLOGI PERMAINAN

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2015



THESIS - TE - 142599

OPTIMIZATION FOR HEROES BEHAVIOURS BASED ON NSGA - II

CINDY TAURUSTA
NRP. 2213205023

DOSEN PEMBIMBING
Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.
Dr. Eko Mulyanto Yuniarno, S.T., M.T.

MAGISTER PROGRAM
EXPERTISE FIELD OF MULTIMEDIA INTELLIGENT NETWORK
CONCENTRATION OF GAME TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
INSTITUTE TECHNOLOGY OF SEPULUH NOPEMBER
SURABAYA
2015

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T.)
di
Institut Teknologi Sepuluh Nopember

Oleh:
Cindy Taurusta
NRP. 2213205023

Tanggal Ujian : 17 Juni 2015
Periode Wisuda : September 2015

Disetujui oleh :

1. Dr. Supeno Mardi Susiki N, S.T., M.T.
NIP. 197003131995121001

(Pembimbing I)

2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.
NIP : 196806011995121009

(Pembimbing II)

3. Dr. I Ketut Eddy Purnama, S.T., M.T.
NIP. 196907301995121001

(Penguji)

4. Dr. Surya Sumpeno, S.T., M.Sc.
NIP. 196906131997021003

(Penguji)

Direktorat Program Pasca Sarjana



Prof. Dr. Ir. Adi Soeprijanto, M.T.
NIP. 196404051990021001

OPTIMASI UNTUK PERILAKU PAHLAWAN BERBASIS NSGA-II

Nama mahasiswa : Cindy Taurusta
NRP : 2213205023
Pembimbing : Dr. Supeno Mardi Susiki Nugroho, S.T. M.T.
Dr. Eko Mulyanto Yuniarno, S.T., M.T.

ABSTRAK

Real Time Strategy merupakan sub genre permainan strategi dimana pemain dapat menang dengan cara menghancurkan instalasi aset bangunan musuh pada operasi penyerangan dan meminimalkan kerusakan pada instalasi dan aset pemain itu sendiri pada operasi pertahanan. Untuk memenangkan *game* RTS ini biasanya terdapat agen terkuat yang disebut dengan Pahlawan. Namun Pahlawan dalam *game* memiliki perilaku yang sama dengan pasukannya, hanya saja ia menjadi agen terkuat. Hal ini yang menyebabkan peran dan fungsi seorang Pahlawan dalam permainan dianggap biasa, tidak unik, dan membosankan, sehingga cenderung ada atau tidaknya Pahlawan dalam game tidak terlalu penting bagi pemain.

Pada penelitian ini merujuk pada pahlawan dalam *game Clash Of Clans* dan penelitian tentang pahlawan dalam *game*. Pahlawan dalam dunia nyata maupun dalam film memiliki beberapa tujuan. Beberapa tujuan ini membuat pahlawan memiliki banyak perilaku. Pemilihan perilaku pahlawan harus dioptimasi agar mendapatkan keuntungan pada operasi penyerangan. Berbasis pada *Non-Dominated Sorting Genetic Algorithm II* (NSGA-II) pencarian solusi terbaik dengan mengoptimalkan 6 parameter penting dari kondisi penyerangan yaitu *health* Pahlawan, *health* musuh, *health* pasukan, *damage received* Pahlawan, *damage received* musuh, dan *damage receive* pasukan, didapatkan optimasi pemilihan perilaku pahlawan sesuai dengan kondisi pertempuran yang dihadapinya.

NSGA-II dapat digunakan untuk memilih perilaku pahlawan secara otomatis. Metode ini berhasil memilih perilaku pahlawan yang sesuai dengan kondisi pertempuran yang dihadapi. Simulasi yang dilakukan pada skenario yang berbeda dapat membuktikan bahwa NSGA-II memberikan hasil yang optimal di setiap skenario tersebut.

Kata Kunci : *Real Time Strategy*, *Hero*, Optimasi , Perilaku Pahlawan, *multi objective*, NSGA-II

Halaman ini Sengaja Dikosongkan

OPTIMIZATION FOR HEROES BEHAVIOR BASED ON NSGA-II

Name of Student : Cindy Taurusta
NRP : 2213205023
Supervisor : Dr. Supeno Mardi Susiki Nugroho, S.T. M.T.
Dr. Eko Mulyanto Yuniarno, S.T., M.T.

ABSTRACT

Real Time Strategy is a sub-genre of strategy game in which players can win by destroying enemy buildings assets installations in offensive operations and minimize damage to the installations and assets of the player it self in defense operations. To win the RTS games, typically there is strongest agent called Hero. But the Hero in the game have the same behavior with his troops, and he just became the strongest agent. This is why the role and function of a Hero in game considered normal, is not unique, and boring, so it tends to presence or absence of Hero in the game is not very important for a player.

In this study refers to the hero in the game Clash Of Clans and research about the hero in the game. Hero in the real world as well as in the film has several purposes. Some of the objectives of this makes the hero has a lot of behavior. Selection of the hero's behavior must be optimized in order to gain advantage in offensive operations. Based on the Non-dominated Sorting Genetic Algorithm II (NSGA-II) searches the best solution to optimize 6 essential parameters of attack, namely health condition hero, enemy health, health of troops, received a hero damage, damage received enemy, and damage receive troops, obtained optimization conduct elections in accordance with the conditions hero battles it faces.

NSGA-II can be used to automatically choose the hero's behavior. This method succeeded in selecting a hero's behavior in accordance with the conditions faced combat. Simulations were performed on different scenarios can prove that the NSGA-II to provide optimal results in each of these scenarios.

Keywords: Real Time Strategy, Hero, Optimization, Heroes Behaviours, multi-objective, NSGA-II

Halaman ini Sengaja Dikosongkan

KATA PENGANTAR



Alhamdulillah, segala puji syukur kepada ALLAH SWT Tuhan Semesta Alam. Berkat rahmat-NYA, penulis dapat menyelesaikan tesis ini dengan baik. Tesis dengan judul “OPTIMASI UNTUK PERILAKU PAHLAWAN MENGGUNAKAN MULTI OBJECTIVE OPTIMIZATION” diselesaikan penulis dalam satu semester, yakni pada semester 4 program Pasca Sarjana ini. Tesis ini disusun guna memenuhi persyaratan untuk memperoleh gelar Magister Teknik pada bidang konsentrasi Teknologi Permainan, bidang studi Jaringan Cerdas Multimedia, jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Keterbatasan kemampuan penulis dalam mengerjakan Tesis ini tidak terlalu menghambat penyelesaian penelitian karena begitu banyak perhatian dan bantuan dari rekan-rekan, para dosen, dan kerabat yang dengan ikhlas meluangkan waktu dan pikirannya untuk membantu penulis. Beberapa pihak yang penulis sebutkan berperan besar dalam penyusunan Tesis ini. Terima kasih penulis ucapkan terutama untuk:

1. *Didik Yuswandono* dan *Surya Ningsih*, Bapak dan Ibu penulis, serta Nenek dan adik – adik penulis yang telah memberikan dukungan moral dan material tanpa henti untuk melalui masa-masa perkuliahan program Magister. Dukungan mereka yang memberikan kekuatan penulis untuk melalui masa-masa sulit.
2. Beasiswa BPPDN DIKTI Calon Dosen yang selain telah membebaskan penulis dari biaya perkuliahan juga memberikan bantuan biaya hidup untuk penulis di jenjang Magister.
3. Dr. Supeno Mardi Susiki Nugroho, S.T., M.T., sebagai Pembimbing I yang dengan sangat sabar membimbing penulis yang terus menerus bertanya meskipun banyak kesalahan dan sering gagal paham “pak, maaf sudah

membuat bapak bosan melihat saya yang bolak balik bertanya, saya sedikit lemot pak”.

4. Dr. Eko Mulyanto Yuniarno, S.T., M.T., sebagai Pembimbing II yang dengan sangat sabar membimbing penulis yang selalu tidak bisa mengutarakan maksud tesisnya sehingga sering gagal paham (juga), “pak nanti saya buat kopi tapi jangan disindir soal nikah ya pak”.
5. Dosen Pengajar Jurusan Teknik Elektro, khususnya bidang keahlian Jaringan Cerdas Multimedia yang telah memberikan pengetahuan dan pengalaman baru kepada penulis.
6. Sahabat terbaik, yang meskipun aku telat menemukannya, aku tak akan melepaskannya, karena aku sangat bersyukur mengenalnya, Vina Oktaviani. Terimakasih untuk semua bantuannya, pengertiannya, perhatiannya, kebersamaan kita selama ini, aku tidak akan pernah lupa, terimakasih banyak.
7. Seseorang yang sangat special di hati saya, di hidup saya, menjadi motivasi dan semangat saya, terimakasih atas kebersamaannya selama ini meskipun sering hubungan gagal paham karena LDR (*Long Distance Relationship*), tapi aku tahu kamu yang terbaik, mas AH ku. Terimakasih juga keluarga kamu.
8. Teman-teman S2 Teknologi Permainan angkatan 2013 yang merupakan “Orang – orang gokil nggak jelas” tapi sangat sangat sangat menyenangkan, *ngangen*, mengesankan sehingga tak henti saya mengucapkan “*You are AMAJING*”. Terima kasih telah menemani selama 2 tahun untuk berbagi ilmu, diskusi cerdas, pengalaman yang telah menikah, dan ilmu – ilmu dewasa lainnya.
9. Sahabat sekelompok dari semester I sampai akhir, Muhammad Aminul Akbar, Raksi Intansari, dan Farid Hartono Gunawan, terimakasih ilmu yang telah kalian bagi selama ini, terimakasih untuk kebersamaan yang tak hanya saat kerja kelompok, tapi semua waktu yang telah kita lalui bersama.
10. Sahabat 7 tahun, sahabat semenjak SMA, di SMAN 9 Surabaya, Leny Saraswati, Madyareta Kashita Susanti, Dwi Okta Aminatus Sholihah, Dwi

Rahmawati, dan Sherly Nindya, terimakasih bantuan, motivasi, dukungan, pengertian, dan kebersamaan kita selama ini.

11. Sahabat Kuliah D3 dan D4 di Politeknik Elektronika Negeri Surabaya, terimakasih untuk setiap ilmu, dukungan, motivasi, dan kebersamaan kita selama ini.
12. Para Supir Angkot P dan G di Surabaya yang selama ini telah mengantarkan penulis dari rumah ke kampus ITS dengan selamat.
13. Kos Gebang Wetan 1A Kamar 11 yang selama ini telah menjadi rumah kedua yang sangat nyaman untukku.

Penulis sepenuhnya menyadari bahwa hasil karya ini sangatlah jauh dari sempurna. Walaupun penulis menganggapnya sebagai pencapaian yang luar biasa tapi tentulah masih banyak kekurangan yang dapat dikoreksi oleh pihak lain. Kritik, saran, maupun studi lebih lanjut dari topik yang penulis sajikan sangat membuat penulis bahagia.

Surabaya, Juni 2015

Penulis.

DAFTAR ISI

ABSTRAK	iii
ABSTRACT	v
KATA PENGANTAR.....	vii
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	Error! Bookmark not defined.
DAFTAR TABEL	Error! Bookmark not defined.
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
BAB 2 DASAR TEORI.....	7
2.1 <i>Game</i>	7
2.1.1 <i>RTS (Real Time Strategy)game</i>	8
2.1.1.1 <i>Game Clash Of Clans</i>	10
2.2 Skenario Dalam <i>Game</i>	10
2.2.1 Skenario <i>Game</i> Non-Perang.....	11
2.2.2 Skenario <i>Game</i> Perang.....	11
2.2.2.1 <i>Game</i> Perang Jarak Jauh.....	11
2.2.2.2 <i>Game</i> Perang Jarak Dekat	12
2.2.2.3 <i>Game</i> Perang Kombinasi.....	12
2.3 Strategi Dalam <i>Game</i>	12
2.3.1 Strategi Menyerang.....	12
2.3.2 Strategi Bertahan.....	13
2.2.1 Skenario Menghindar	13
2.4 Operasi Penyerangan.....	14
2.5 Optimasi	14
2.5.1 Teknik Optimasi.....	15
2.5.2 Metode Optimasi.....	16
2.6 Perilaku	17
2.7 Pahlawan	17

2.8	Perilaku Pahlawan	18
2.9	<i>Big Five Personality</i> (Lima Besar Kepribadian)	19
2.9.1	Penelitian <i>Big Five Personality</i>	19
2.9.2	Pikiran Akhir	20
2.10	<i>Agresif</i> (Penyerang)	21
2.10.1	Karakteristik Komunikasi <i>Agresif</i>	21
2.10.2	Definisi Kerja Agresi [25]	23
2.10.3	Bentuk Agresi	24
2.10.4	Agresi Proaktif dan Reaktif	24
2.10.5	Pembelajaran Sosial	25
2.11	<i>Supportif</i> (Pendukung atau Pelindung) [26]	28
2.12	<i>Leadership</i> (Kepemimpinan) [26]	28
2.12.1	Kepemimpinan dan Karakteristik Pribadi	30
2.12.2	Beberapa Definisi Kepemimpinan	30
2.12.3	Prinsip – prinsip Kepemimpinan	32
2.12.4	Kepemimpinan yang Dinamis	32
2.12.5	Gaya Kepemimpinan	34
2.13	Egois	37
2.14	<i>Multi Objective Optimization</i>	37
2.15	Pareto Optimal	39
2.16	Algoritma Genetika	41
2.16.1	Pengertian Algoritma Genetika [30]	41
2.16.2	Struktur Umum Algoritma Genetika	42
2.16.3	Pengkodean	45
2.16.3.1	Pengkodean Biner	45
2.16.3.2	Pengkodean Bilangan Riil atau Pengkodean Permutasi	45
2.16.3.3	Pengkodean Nilai	46
2.16.3.4	Pengkodean Struktur Data atau Pengkodean Pohon ...	46
2.16.4	Fungsi Evaluasi Kebugaran (<i>Fitness</i>)	46
2.16.5	Operator Genetik	47
2.16.5.1	Seleksi	47
2.16.5.2	<i>Crossover</i>	49
2.16.5.3	Mutasi	50
2.16.5.4	Elitisme	53

2.15.5.5	Parameter Algoritma Genetika	53
2.17	Algoritma NSGA-II	55
2.17.1	Inisialisasi Populasi.....	58
2.17.2	<i>Non-Dominated Sort</i>	58
2.17.3	<i>Crowding Distance</i>	59
2.17.4	<i>Selection</i>	59
2.17.5	<i>Genetic Operator</i>	60
2.17.6	<i>Recombination</i> dan Seleksi	60
2.18	<i>Finite State Machine</i> (FSM).....	60
BAB 3	METODOLOGI PENELITIAN.....	63
3.1	Lingkup Penelitian	64
3.2	Deskripsi Permainan	65
3.3	Tahapan Pembuatan Sistem	66
3.4	Desain 3 Pahlawan DWIPA YUDHA.....	67
3.4.1	Desain Objek 3 Dimensi (3D) Pahlawan	67
3.4.2	Desain Karakteristik Pahlawan	68
3.4.3	Parameter Diri Setiap Pahlawan	70
3.5	Lima Perilaku Dasar Pahlawan	72
3.6	Desain Skenario Pahlawan dan Pasukan Terhadap Musuh.....	75
3.7	Optimasi Multi Objective.....	86
3.7.1	Populasi Awal	87
3.7.2	Evaluasi Fungsi Objective	87
3.7.3	Pengurutan Non-Dominasi (<i>Non-dominated Sort</i>)	90
3.7.4	<i>Crowding Distance</i>	91
3.7.5	<i>Offspring</i> (Keturunan).....	92
3.7.5.1	Seleksi.....	92
3.7.5.2	<i>Crossover</i> (Persilangan)	92
3.7.5.3	Mutasi.....	93
3.8	Skenario Pengujian.....	94
BAB 4	HASIL DAN PEMBAHASAN.....	95
4.1	Pengujian Performa NSGA-II.....	95
4.1.1	Pengujian 2 Fungsi Objective Batara 100 Populasi 20 Generasi	96
4.1.2	Pengujian 2 Fungsi Objective Batara 100 Populasi 50 Generasi	98

4.1.3	Pengujian 2 Fungsi Objective Batara 100 Populasi 75 Generasi	101
4.1.4	Pengujian 2 Fungsi Objective Batara 100Populasi 100Generasi	104
4.1.5	Pengujian 2 Fungsi ObjectiveWira 100 Populasi 20 Generasi	110
4.1.6	Pengujian 2 Fungsi Objective Wira 100 Populasi 50 Generasi	113
4.1.7	Pengujian 2 Fungsi Objective Wira 100 Populasi 75 Generasi	116
4.1.8	Pengujian 2 Fungsi Objective Wira 100 Populasi 100 Generasi	119
4.1.9	Pengujian 2 Fungsi Objective Kirna 100 Populasi 20 Generasi	126
4.1.10	Pengujian 2 Fungsi Objective Kirna 100 Populasi 50 Generasi	129
4.1.11	Pengujian 2 Fungsi Objective Kirna 100 Populasi 75 Generasi	133
4.1.12	Pengujian 2 Fungsi Objective Kirna 100 Populasi 100 Generasi	135
4.2	Simulasi Pada Permainan	142
BAB 5	KESIMPULAN.....	151
5.1	Kesimpulan	151
5.2	Saran	153
	DAFTAR PUSTAKA	155
	LAMPIRAN	
	BIOGRAFI PENULIS.....	

DAFTAR TABEL

Tabel 2.1 Tabel <i>genre game</i>	7
Tabel 2.2 Contoh Pengkodean Biner	45
Tabel 2.3 Contoh Pengkodean Bilaang Riil atau Bilangan Permutasi.....	45
Tabel 2.4 Contoh Pengkodean Nilai	46
Tabel 2.5 Contoh Pengkodean Cabang	46
Tabel 2.6 Tabel <i>Crossover</i> 1-Titik.....	49
Tabel 2.7 Tabel <i>Crossover</i> 2-Titik.....	50
Tabel 2.8 Tabel <i>Crossover</i> Seragam	50
Tabel 2.9 Contoh Mutasi pada Pengkodean Biner.....	51
Tabel 2.10 Contoh Mutasi pada Pengkodean Permutasi.....	51
Tabel 3.1 Karakteristik Pahlawan DWIPA YUDHA.....	70
Tabel 3.2 Rincian Parameter Batara Maheswara	71
Tabel 3.3 Rincian Parameter Wira Oragastra	71
Tabel 3.4 Rincian Parameter Kirna Waranggani	72
Tabel 3.5 Tabel Kondisi Kesehatan Pahlawan dan Pasukan	79
Tabel 3.6 Tabel Kombinasi Kesehatan Pahlawan + Pasukan	81
Tabel 3.7 Tabel <i>Damage Receive</i> Pahlawan dan Pasukan.....	82
Tabel 3.8 Tabel Kesehatan Musuh.....	83
Tabel 4.1 Tabel Dua Fungsi Objective Batara 100 Populasi 20 Generasi	96
Tabel 4.2 Tabel Dua Fungsi Objective Batara 100 Populasi 50 Generasi.....	98
Tabel 4.3 Tabel Dua Fungsi Objective Batara 100 Populasi 75 Generasi.....	101
Tabel 4.4 Tabel Dua Fungsi Objective Batara 100 Populasi 100 Generasi	104
Tabel 4.5 Hasil NSGA-II Lima Perilaku Batara 100 Populasi 100 Generasi	109
Tabel 4.6 Tabel Dua Fungsi Objective Wira 100 Populasi 20 Generasi	110
Tabel 4.7 Tabel Dua Fungsi Objective Wira 100 Populasi 50 Generasi	113
Tabel 4.8 Tabel Dua Fungsi Objective Wira 100 Populasi 75 Generasi	116
Tabel 4.9 Tabel Dua Fungsi Objective Wira 100 Populasi 100 Generasi	119
Tabel 4.10 Hasil NSGA-II Lima Perilaku Wira 100 Populasi 100 Generasi	125
Tabel 4.11 Tabel Dua Fungsi Objective Kirna 100 Populasi 20 Generasi	126
Tabel 4.12 Tabel Dua Fungsi Objective Kirna 100 Populasi 50 Generasi	129
Tabel 4.13 Tabel Dua Fungsi Objective Kirna 100 Populasi 75 Generasi	132
Tabel 4.14 Tabel Dua Fungsi Objective Kirna 100 Populasi 100 Generasi	135
Tabel 4.15 Hasil NSGA-II Lima Perilaku Kirna 100 Populasi 100 Generasi	139
Tabel 4.16 Tabel hasil penyerangan dengan perilaku optimal <i>agresif</i>	143
Tabel 4.17 Tabel hasil penyerangan biasa dengan kondisi sama dengan <i>agresif</i>	144
Tabel 4.18 Tabel hasil penyerangan dengan perilaku optimal <i>supportif</i>	146
Tabel 4.19 Tabel hasil penyerangan biasa dengan kondisi sama dengan <i>supportif</i>	147

Tabel 4.20 Tabel hasil penyerangan dengan perilaku optimal <i>leadership</i>	149
Tabel 4.21 Tabel hasil penyerangan biasa dengan kondisi sama dengan <i>leadership</i>	150

DAFTAR GAMBAR

Gambar 1.1 Pahlawan Dalam Penelitian.....	4
Gambar 2.1 <i>Game</i> RTS <i>Age Of Empires</i> dan <i>Starcraft</i>	9
Gambar 2.2 Arsitektur kecerdasan buatan untuk RTS <i>Game</i>	9
Gambar 2.3 Referensi <i>Game</i> RTS untuk Penelitian : <i>Clash Of Clans</i>	10
Gambar 2.4 Minimum dari $f(x)$ sama dengan Maksimum dari $-f(x)$	15
Gambar 2.5 Grafik <i>Big Five Personality</i>	20
Gambar 2.6 Diagram Empat Gaya Kepemimpinan	36
Gambar 2.7 Konsep Pareto Optimal	40
Gambar 2.8 Diagram Alur Algoritma Genetika.....	43
Gambar 2.9 Ilustrasi Operator Dengan Satu Titik Persilangan.....	49
Gambar 2.10 Ilustrasi Operator Dengan Dua Titik Persilangan	50
Gambar 2.11 Ruang Keputusan dan Obyektif (<i>Decision and Objective Space</i>) ...	57
Gambar 2.12 Blok Diagram NSGA-II [34]	57
Gambar 2.13 Contoh FSM	61
Gambar 3.1 Diagram Ven Lingkup Penelitian	64
Gambar 3.2 Pohon Penelitian	65
Gambar 3.3 Diagram <i>Fishbone</i> Optimasi Perilaku Pahlawan	66
Gambar 3.4 Desain 3D pahlawan DWIPA YUDHA.....	67
Gambar 3.5 FSM Pahlawan dalam permainan	76
Gambar 3.6 <i>Rule Base</i> Jumlah pasukan setiap Townhall	61
Gambar 4.1 <i>Pareto Optimal</i> Batara dengan 100 Populasi 20 Generasi.....	97
Gambar 4.2 Pembagian 4 kuadran <i>pareto front</i> 100 populasi 20 generasi	97
Gambar 4.3 <i>Pareto Optimal</i> Batara dengan 100 Populasi 50 Generasi	99
Gambar 4.4 Pembagian 4 kuadran <i>pareto front</i> 100 populasi 50 generasi	100
Gambar 4.5 <i>Pareto Optimal</i> Batara dengan 100 Populasi 75 Generasi.....	102
Gambar 4.6 Pembagian 4 kuadran <i>pareto front</i> 100 populasi 75 generasi	103
Gambar 4.7 <i>Pareto Optimal</i> Batara dengan 100 Populasi 100 Generasi.....	105
Gambar 4.8 Pembagian 4 kuadran <i>pareto front</i> 100 populasi 100 generasi	106
Gambar 4.9 Perbandingan setiap generasi pada 100 populasi	107
Gambar 4.10 Grafik radar plot kelima perilaku Batara Maheswara	110
Gambar 4.11 <i>Pareto Optimal</i> Wira dengan 100 Populasi 20 Generasi	111
Gambar 4.12 Pembagian 4 kuadran <i>pareto front</i> 100 populasi 20 generasi	112
Gambar 4.13 <i>Pareto Optimal</i> Wira dengan 100 Populasi 50 Generasi	114
Gambar 4.14 Pembagian 4 kuadran <i>pareto front</i> 100 populasi 50 generasi	115
Gambar 4.15 <i>Pareto Optimal</i> Wira dengan 100 Populasi 75 Generasi	117
Gambar 4.16 Pembagian 4 kuadran <i>pareto front</i> 100 populasi 75 generasi	118
Gambar 4.17 <i>Pareto Optimal</i> Wira dengan 100 Populasi 100 Generasi	120

Gambar 4.18 Pembagian 4 kuadran pareto front 100 populasi 100 generasi	121
Gambar 4.19 Perbandingan setiap generasi pada 100 populasi.....	122
Gambar 4.20 Grafik radar plot kelima perilaku Wira Oragastra	126
Gambar 4.21 <i>Pareto Optimal</i> Kirna dengan 100 Populasi 20 Generasi.....	127
Gambar 4.22 Pembagian 4 kuadran <i>pareto front</i> 100 populasi 20 generasi.....	128
Gambar 4.23 <i>Pareto Optimal</i> Kirna dengan 100 Populasi 50 Generasi	130
Gambar 4.24 Pembagian 4 kuadran pareto front 100 populasi 50 generasi	131
Gambar 4.25 <i>Pareto Optimal</i> Kirna dengan 100 Populasi 75 Generasi.....	133
Gambar 4.26 Pembagian 4 kuadran pareto front 100 populasi 75 generasi	134
Gambar 4.27 <i>Pareto Optimal</i> Kirna dengan 100 Populasi 100 Generasi.....	136
Gambar 4.28 Pembagian 4 kuadran pareto front 100 populasi 100 generasi	137
Gambar 4.29 Perbandingan setiap generasi pada 100 populasi.....	138
Gambar 4.30 Grafik radar plot kelima perilaku Kirna Waranggani.....	142
Gambar 4.31 Tampilan Simulasi awal dan akhir perilaku <i>agresif</i> pahlawan	143
Gambar 4.32 Perilaku Pahlawan tanpa optimasi dengan kondisi yang sama dengan perilaku <i>agresif</i>	144
Gambar 4.33 Tampilan Simulasi awal dan akhir perilaku <i>supportif</i> pahlawan ..	145
Gambar 4.34 Perilaku Pahlawan tanpa optimasi dengan kondisi yang sama dengan perilaku <i>supportif</i>	147
Gambar 4.35 Tampilan Simulasi awal dan akhir perilaku <i>leadership</i> pahlawan	148
Gambar 4.36 Perilaku Pahlawan tanpa optimasi dengan kondisi yang sama dengan perilaku <i>leadership</i>	149

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teori permainan merupakan sebuah sistem matematika yang digunakan untuk menganalisa dan memprediksi bagaimana manusia berperilaku dalam situasi yang strategis. Analisa standart keseimbangan menganggap bahwa semua pemain : (1) membentuk kepercayaan berdasarkan sebuah analisa dari apa yang kekuatan lain lakukan (strategi berpikir); memilih respon terbaik yang memberikan kepercayaan (optimisasi); dan mengatur respon terbaik dan kepercayaan sampai konsisten mereka berubah – ubah (keseimbangan) [1]. Ada banyak jenis permainan di dunia, salah satu yang saat ini sedang dikembangkan dan sangat terkenal yaitu jenis RTS (Real Time Strategy). Game RTS sering menggunakan dua “level” dari AI [2] : satu diwakili dengan sebuah NPC, membuat keputusan pada sekumpulan kesatuan (pekerja, pasukan, senjata, kendaraan atau bangunan), dan satu yang lainnya adalah akrab pada setiap kesatuan kecil [3]. Dalam penelitian ini menggunakan Game RTS yaitu DWIPA YUDHA, game yang dibuat oleh sesama Mahasiswa Pascasarjana Game Teknologi ITS 2013. Dalam game DWIPA YUDHA ini terdiri dari 2 operasi yaitu operasi menyerang dan operasi bertahan.

Dalam operasi menyerang, diperlukan strategi yang tepat agar serangan yang dilakukan untuk mengalahkan tim lawan menjadi optimal. Keoptimalan dari strategi menyerang dapat diukur dari sedikit banyaknya musuh yang tewas & luka – luka atau diukur dari jumlah kesehatan yang tersisa dari setiap anggota/musuh [4]. Pada game RTS yang lain yaitu Strike Ops, pemain dapat menang dengan cara menghancurkan instalasi aset bangunan musuh pada operasi penyerangan dan meminimalkan kerusakan pada instalasi dan aset pemain itu sendiri pada operasi pertahanan [5]. Sama seperti *game RTS* pada umumnya, operasi penyerangan *game* DWIPA YUDHA juga dapat dimenangkan dengan menghancurkan instalasi aset bangunan musuh. Untuk mendukung aksi serangan tersebut, pemain memiliki beberapa aset pendukung, seperti unit pasukan dan pahlawan yang akan disebar di

setiap daerah pada peta, senjatanya, perilakunya, serta keahlian – keahlian khusus (*ability*) yang dimiliki oleh masing – masing pasukan dan pahlawan untuk menyerang dan menghancurkan target atau musuhnya.

Dari beberapa komponen dalam *game* DWIPA YUDHA tersebut, ada satu hal yang lebih ditekankan untuk penelitian ini, yaitu optimasi perilaku pahlawan. Pada penelitian lain [6] seorang pahlawan atau boss dalam sebuah *game* hanya memiliki satu perilaku yaitu terus menerus menyerang demi memenangkan permainan (*agresif*). Perilaku yang dimiliki sama dengan pasukannya, perbedaannya hanya ia menjadi agen terkuat atau memiliki nyawa yang lebih besar dari pasukannya. Maka pada penelitian ini, diciptakanlah pahlawan yang berbeda dengan pasukannya, tidak hanya memiliki nyawa lebih besar atau terkuat dan memiliki satu perilaku yang hanya terus menerus menyerang, namun ia juga akan memiliki 5 perilaku yang berbeda yang saling bertentangan. Kelima perilaku inipun juga akan dioptimasi, sehingga akan terjadi variasi munculnya kelima perilaku pahlawan tersebut saat menemui berbagai macam musuh.

Dalam Kamus Bahasa Indonesia, W.J.S. poerdwadarminta (1997 :753) dikemukakan bahwa : “Optimasi adalah hasil yang dicapai sesuai dengan keinginan, jadi optimasi merupakan pencapaian hasil sesuai harapan secara efektif dan efisien”. Optimasi juga banyak diartikan sebagai ukuran dimana semua kebutuhan dapat dipenuhi dari kegiatan-kegiatan yang dilaksanakan [7]. Menurut Winardi (1999 : 363) Optimasi adalah ukuran yang menyebabkan tercapainya tujuan sedangkan jika dipandang dari sudut usaha, optimasi adalah usaha memaksimalkan kegiatan sehingga mewujudkan keuntungan yang diinginkan atau dikehendaki. Dari uraian tersebut diketahui bahwa optimasi hanya dapat diwujudkan apabila dalam pewujudannya secara efektif dan efisien [8]. Optimasi pada penelitian ini adalah bagaimana membuat pahlawan dalam *game* berperilaku seperti pahlawan yang kita tahu dalam dunia nyata maupun dalam film.

Bila pahlawan dalam dunia nyata ketika dihadapkan dengan masalah bahwa ia harus menyelamatkan orang – orang yang lemah dari musuh yang sangat kuat, maka ia harus menjadi seorang pelindung meskipun risikonya sendiri ialah ia harus mati, namun bila ia sudah mengetahui bahwa musuhnya tersebut dalam keadaan lemah, maka ia memiliki kesempatan untuk menyerang musuhnya sampai mati

sehingga dapat dikatakan ia telah memenangkan pertempuran. Kedua perilaku tersebut bertentangan karena harus mengorbankan salah satu, yaitu antara pahlawan yang mati atau orang yang lemah tersebut yang mati (tidak terlindungi). Maka pahlawan dalam *game* sebaiknya juga mengasumsi perilaku pahlawan dalam dunia nyata tersebut namun tidak harus mengorbankan salah satu antara nyawa pahlawan atau nyawa yang lemah, sehingga masih dapat memenangkan pertempuran.

Berdasarkan yang diinginkan tersebut, pahlawan dalam DWIPA YUDHA tidak hanya dapat mengoptimasi 2 perilaku yang saling bertentangan, yaitu antara menyelamatkan yang lemah dengan resiko ia mati atau hanya fokus menyerang musuh untuk memenangkan pertempuran dengan resiko orang yang lemah mati, namun juga pahlawan DWIPA YUDHA dapat memiliki 5 perilaku yang berbeda yang saling bertentangan, yaitu *agresif* yang terus menerus menyerang menghancurkan instalasi musuh, *supportif* yang melindungi pasukannya yang lemah, *leadership* yang memimpin pertempuran, *egois 1* yang melarikan diri, dan *egois 2* yang melarikan diri dengan mengajak pasukannya melarikan diri bersamanya, dimana meskipun kelima perilaku tersebut bertentangan, pahlawan tetap dapat memenangkan pertempuran tanpa banyak mengorbankan nyawanya dan nyawa pasukannya.

Perilaku tambahan selain *agresif* dan *supportif* adalah *leadership*. Seperti kita ketahui bahwa pahlawan pasti memiliki jiwa dan sikap kepemimpinan. Dari tahun 1950 sampai tahun 1970-an, cara menonjol untuk konsep perilaku kepemimpinan adalah dalam hal kepedulian terhadap tujuan tugas dan terhadap kepedulian bagi orang-orang (atau hubungan). Teori-teori baru menekankan emosi dan nilai-nilai yang diperlukan untuk memahami bagaimana seorang pemimpin dapat mempengaruhi pengikutnya (pasukannya) untuk mengorbankan diri, berkomitmen untuk tujuan ideologis, dan mencapai lebih dari kemungkinan awal yang mereka percaya, dan juga teori – teori baru tersebut mengakui pentingnya perilaku simbolis dan peran pemimpin dalam membuat sesuatu yang berarti bagi pengikutnya (pasukannya) [9]. Mengoptimasi dua atau lebih keadaan berbeda yang saling bertentangan inilah disebut dengan *Multi Objective Optimization*.

Multi Objective Optimization (MOO) adalah suatu proses yang secara simultan mengoptimalkan dua atau lebih fungsi tujuan yang saling

bertentangan dengan kendala yang ada (Gutierrez, 2012). Multi-objective Optimization dalam kasus nyata banyak digunakan untuk menangani permasalahan yang harus memenuhi lebih dari satu fungsi tujuan [10]. Untuk mengoptimasi MOO pada penelitian ini digunakan *Non-Domination Sorted Genetic Algorithm – II* (NSGA – II).

NSGA-II merupakan pengembangan dari Algoritma Genetika (GA) kemudian NSGA. GA berbasis populasi, beberapa solusi dapat disimpan secara serentak di seluruh proses. GA hanya mampu menangani masalah optimasi satu objective sedangkan NSGA mampu menangani masalah optimasi multiobjective di mana solusi berbeda dapat ditemukan berdasarkan pada gagasan non-dominasi. Sebuah solusi disebut non-didominasi jika itu tidak didominasi oleh solusi lain. Tujuh tahun setelah Deb dkk [11] memperkenalkan versi baru: NSGA-II, di mana solusi konvergensi dan penyebaran ditingkatkan. Para penulis memvalidasi algoritma mereka menggunakan sembilan pengujian masalah. Hasil pengujian menunjukkan bahwa NSGA-II menyajikan hasil yang lebih baik dan dukungan diversity dibandingkan dengan dua MOEAs elitis lainnya, Paes dan SPEA [12]. Berikut ini 3 jenis pahlawan DWIPA YUDHA dalam penelitian ini :



(a)

(b)

(c)

Gambar 1.2 Pahlawan dalam penelitian (a) Batara Maheswara (b) Wira Oragatra (c) Kirna Waranggani

1.2 Rumusan Masalah

Pahlawan yang hanya memiliki satu tujuan yaitu rela mati untuk memenangkan pertandingan, membuat permainan menjadi membosankan dan membuat peran pahlawan tak penting pahlawan yang kita tahu di dunia nyata atau pada film. Dalam sebuah *game*, pahlawan bukanlah pasukan sehingga peran dan tujuannya harus dibedakan. Pahlawan yang hanya dapat berperang sendiri membuatnya mudah dikalahkan sehingga membuat *game* tersebut menjadi kurang menarik.

1.3 Batasan Masalah

Pada penelitian ini perilaku pahlawan harus lebih dari satu dan pengoptimasian perilaku pahlawan tersebut hanya pada operasi penyerangan.

1.4 Tujuan Penelitian

Mendapatkan perilaku pahlawan yang optimal sesuai kondisi pertempuran dari beberapa pilihan perilaku yang ada berbasis pada NSGA-II.

1.5 Manfaat Penelitian

Dapat memaksimalkan strategi yang digunakan pemain ketika menyerang lawan sehingga memungkinkan pemain mendapatkan hasil penyerangan yang maksimal yang dapat dicapai.

Halaman ini Sengaja Dikosongkan

BAB 2

DASAR TEORI

Untuk memberikan gambaran secara umum, bab ini akan membahas secara lengkap mengenai seluruh teori yang mendukung penelitian ini mulai dari *Game*, *Game RTS*, *Game Clash Of Clans*, Optimasi, beberapa Perilaku Pahlawan, hingga metode yang digunakan yaitu *Non-Dominated Sorted Genetic Algorithm-II* (NSGA-II).

2.1 *Game*

Pada umumnya *game* digolongkan menjadi beberapa *genre* [13], diantaranya *RPG*, *Adventure game*, *RTS*, *FTPS*, *platform games*, *shooter games*, *sport games*, *racing classic games*, *fighting games* [14], setiap *genre* akan dijabarkan dengan penjelasan beserta contohnya berikut ini :

Tabel 2.1 Tabel *genre game*

NO	Genre Game	Contoh
1	RPG (Role Play Games)	<i>Ragnarok</i> , <i>Diablo</i> , dan <i>Final Fantasy</i>
2	Adventure Game	<i>The Longest Journey</i> , <i>Syberia I</i> , <i>Syberia II</i> .
3	RTS (Real Time Strategy)	<i>Age of Empire</i> (semua versi), <i>Empire Earth</i> (I, II, dan III), <i>Starcraft II</i> , <i>Clash Of Clans</i>
4	FPS (First Person Shooter)	<i>Half Life</i> dan <i>Counter Strike</i> .
5	Platform Games	Bervariasi, beberapa mekanisme lompatan dimodifikasi dengan bantuan alat – alat tertentu seperti tali pengayun dengan panjang yang telah diatur, atau melompat dari trampoline
6	TPS (Third Person Shooter)	<i>GTA</i> , <i>Tomb Raider</i> , <i>Global Ops</i> , <i>Max Payne</i> , <i>IronMan</i>
7	Fighting Games	<i>Tekken 3</i> dan <i>Street Fighter</i>
8	Board Games	<i>chess</i> , <i>mahjong</i> , <i>Sudoku</i> , <i>tic tac toe</i> ,

		<i>card game, monopoly, shake and ladder</i>
9	Puzzle Games	<i>Tetris, Bust A Move, Bejeweled. Solusi dari puzzle</i>
10	Simulation Games	<i>SimCity, Flight simulators</i>
11	Serious Games	<i>Heifer Village: Nepal, MaxDebat</i>
12	Sport Games	<i>Madden NFL, Tiger Wood, Pool 3D, Evolution Soccer, NBA Live, NCAA Football.</i>
13	Web Based Games	<i>Super Mario bross flash, Farmville, Mafia War</i>
14	Massively Multiplayer Online (MMO)	<i>Lineage, WoW (World Of Warcraft), Diablo</i>

2.1.1 RTS (*Real Time Strategy*) game

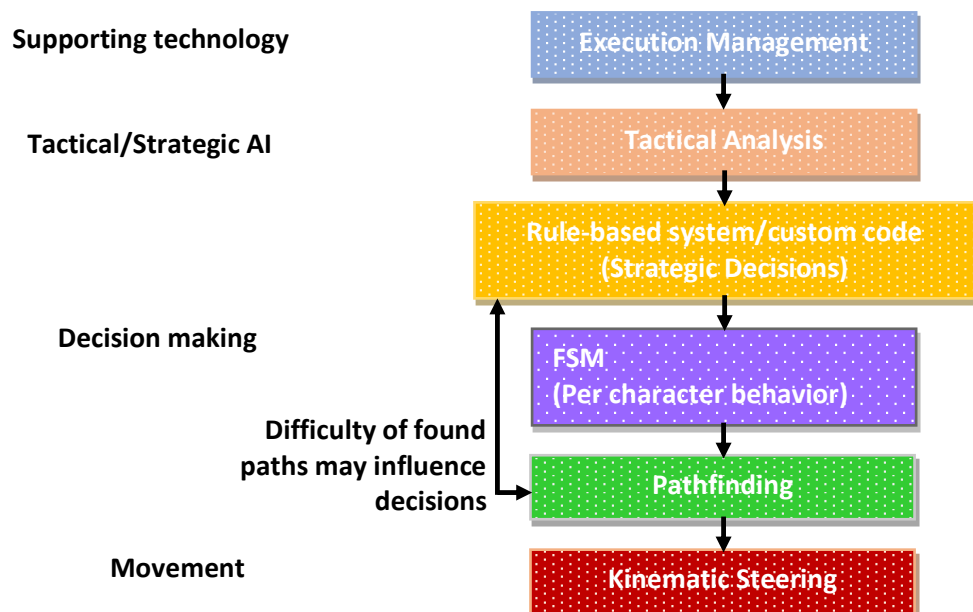
Real Time Strategy Games atau biasa disingkat dengan *RTS Games* adalah [15] *Dumell* yang dibuat oleh *Westwod Studio* pada tahun 1992 yang kemudian *genre* RTS menjadi banyak digemari dan mulai bermunculan *game* untuk dimainkan di *Personal Computer* (PC) lain dengan *genre* yang sama (Millington, 2009). Pada *game* RTS pada umumnya kecerdasan buatan yang digunakan adalah *pathfinding*, *group movement*, *tactical and strategic AI*, dan *decision making*. Strategi real-time (RTS) adalah sub-genre strategi video game yang tidak berkembang secara bertahap dan bergantian. Dalam RTS, seperti dalam game pertempuran lain, para peserta posisi dan *control unit* serta struktur di bawah kendali mereka untuk mengamankan wilayah dari peta dan / atau merusak kontrol lawan mereka. Yang khas dari RTS adalah dalam membuat unit dan struktur tambahan selama permainan. Hal ini umumnya dibatasi oleh persyaratan untuk mengeluarkan akumulasi sumber daya. Sumber daya ini pada dikumpulkan dengan mengendalikan titik-titik khusus pada peta dan / atau memiliki beberapa jenis unit dan struktur yang ditujukan untuk tujuan ini. Lebih khususnya, permainan khas dalam permainan berjenis RTS ini terdapat fitur pengumpulan sumber daya, bangunan dasar, pengembangan teknologi dalam game dan kontrol langsung dari unit. Tugas seorang pemain untuk berhasil pada RTS bisa sangat menuntut, dan *user interface* yang kompleks telah berevolusi untuk mengatasi tantangan.

Beberapa fitur telah dipinjam dari lingkungan desktop ; misalnya, teknik “mengklik dan menyeret” untuk memilih semua unit di bawah daerah tertentu. Game berjenis inilah yang saya gunakan untuk mensimulasikan hasil penelitian. Contoh *game* RTS adalah *Age of Empire* (semua versi), *Empire Earth* (I, II, dan III), *Starcraft II*, *Clash Of Clans*.



Gambar 2.1 *Game RTS Age Of Empires dan Starcraft*

Pada Gambar 2.4 merupakan arsitektur kecerdasan buatan untuk RTS *game*. Variasi dari arsitektur dapat berbeda tergantung dari *gameplay* masing – masing *game*, namun model dari arsitektur kecerdasan buatan untuk RTS *game* pada diagram dibawah ini dapat digunakan sebagai strating point untuk awalan dalam membuat game RTS (Millington, 2009).



Gambar 2.2 Arsitektur kecerdasan buatan untuk RTS *Game*

2.1.1.1 *Game Clash Of Clan*

Clash of Clans adalah permainan strategi di mana pemain dapat membangun dan memperluas desa sendiri, membuka berturut-turut prajurit dan pertahanan yang lebih kuat, serangan dan penjarahan sumber daya dari desa lain, membuat dan bergabung dengan klan dan masih banyak lagi. Pada penelitian ini, dikhususkan untuk meneliti perilaku pahlawan, seperti dalam *Clash Of Clans* yaitu Barbarian King dan Archer Queen. Pahlawan adalah tentara yang paling kuat dalam permainan. Mereka abadi, yang berarti Anda hanya perlu melatih mereka sekali. Namun, jika mereka telah dikalahkan mereka harus regenerasi untuk melawan lagi. Bila pada referensi *game Clash of Clans* terdapat 2 pahlawan yaitu Barbarian King dan Archer Queen, kini akan ada tiga jenis pahlawan dalam game yang menyerupai *Clash Of Clans* yaitu DWIPA YUDHA, yaitu Batara Maheswara, Wira Oragasta, dan Kirna Warangani. Bila pahlawan pada *Clash Of Clans* yaitu Barbarian King maupun Archer Queen menunjukkan bahwa perilakunya hanya bisa melakukan penyerangan dan berdiam diri dalam altarnya ketika kesehatannya habis, maka pada penelitian ini menunjukkan bahwa perilaku pahlawan dapat dioptimasi yaitu tidak hanya bisa melakukan penyerangan (*Offensive*) untuk mendapatkan harta musuh sebanyak – banyaknya atau menghancurkan bangunan musuh sebanyak banyaknya, namun pahlawan dalam penelitian ini juga dapat melindungi pasukannya yaitu memiliki perilaku *Leadership*.



Gambar 2.3 Referensi *Game RTS* untuk penelitian : *Clash Of Clans*

2.2 Skenario Dalam *Game*

Skenario/konsep permainan yang melatar belakangi suatu game secara garis besar dapat dibagi menjadi 2 bagian [13] :

1. Skenario game Non-Perang
2. Skenario game Perang

2.2.1 Skenario *Game* Non-Perang

Game yang termasuk dalam skenario Non-Perang kebanyakan game yang mengandalkan Ketangkasan, Kecermatan, & Kecepatan dan tidak berlatar belakang pertempuran/pertarungan seperti game yang bertipe *Sport Game*, *Racing Game*, *Puzzle Game*, *Board Game*.

2.2.2 Skenario *Game* Perang

Skenario pada game ini berlatar belakang pertempuran/pertarungan baik perang seperti yang ada di dunia nyata maupun perang yang fiksi. Adegan perang dalam game tersebut selalu melibatkan dua/lebih pihak yang bertikai. Dalam penelitian ini menggunakan scenario game Perang, karena game DWIPA YUDHA dilatar belakangi oleh sebuah peperangan dalam merebut kekuasaan atas seluruh wilayah. Secara umum skenario game perang dibagi menjadi 3 jenis berdasarkan jarak pertempuran :

1. Game Perang Jarak Jauh
2. Game Perang Jarak Dekat
3. Game Perang Kombinasi

2.2.2.1 *Game* Perang Jarak Jauh

Skenario game perang yang berjenis pertarungan jarak jauh (Range Combat) ini dalam menyerang & mengalahkan lawan menggunakan senjata/kesaktian sihir dengan jarak serangan yang jauh. Dimana kemampuan serangan dari skenario game ini tergantung dari jumlah amunisi jika menggunakan senjata & kekuatan ilmu (manna) jika menggunakan kesaktian sihir. *Game* perang jarak jauh digunakan dalam penelitian khususnya untuk Pahlawan dan Pasukan yang bersenjatakan panah, karena untuk menyerang menggunakan panah, NPC tersebut dapat membidik musuhnya dari jarak jauh. Contoh : *Starcraft*, *Command&Conquer*, *Waizone*, *WWII*, *Sudden Death*

2.2.2.2 Game Perang Jarak Dekat

Game perang yang termasuk ke dalam jenis pertarungan jarak dekat (melee combat) dalam menyerang & mengalahkan lawan dapat menggunakan tangan kosong dengan pukulan maupun menggunakan senjata dengan jarak yang pendek seperti pedang, kapak, pisau, tongkat. *Game* perang jarak dekat digunakan dalam penelitian khususnya untuk Pahlawan dan Pasukan yang bersenjatakan pedang atau clurit, karena untuk menyerang menggunakan pedang atau clurit, NPC tersebut harus mendekati musuhnya. Contoh : *skyrim 5, zeno dash, arcade games, boxing games, WWF Wrestlemania, Gladiators, Glorius Combat*.

2.2.2.3 Game Perang Kombinasi

Skenario game perang berjenis pertarungan kombinasi merupakan gabungan dari dua skenario game pertarungan yaitu skenario game pertarungan jarak jauh dan jarak dekat. NPC dalam game berjenis ini tidak hanya mempunyai kemampuan bertarung dalam jarak dekat atau jarak jauh saja, tetapi juga dapat mempunyai kemampuan keduanya, bergantung karakter yang ada dalam game tersebut. *Game* perang kombinasi antara jarak dekat dan jarak jauh digunakan dalam penelitian khususnya untuk Pahlawan dan Pasukan yang bersenjatakan sihir, karena untuk menyerang menggunakan sihir, NPC tersebut dapat membunuh musuhnya dari jarak dekat maupun jarak jauh. Contoh : *warcraft, age of empire, age of mythodology, heroes of newwerth*

2.3 Strategi dalam Game

Terdapat 3 macam strategi dalam sebuah game, yaitu [4] :

1. Strategi Menyerang
2. Strategi Bertahan
3. Strategi Menghindar

2.3.1 Strategi Menyerang

Menyerang secara berkelompok dengan koordinasi pada suatu pertempuran dapat menghasilkan strategi penyerangan yang optimal. Keoptimalan dari strategi menyerang dapat diukur dari banyak sedikitnya korban yang tewas dan luka – luka

atau diukur dari jumlah *health* yang tersisa dari setiap anggota kelompok. Metode penyerangan dapat bervariasi bergantung dari tipe pertempuran itu sendiri. Tipe pertempuran jarak jauh hanya mempertimbangkan jumlah amunisi serta jarak penyerangan. Untuk tipe pertempuran jarak dekat harus dipertimbangkan kekuatan serangan dan pemilihan lawan yang tepat supaya strategi yang dilakukan menjadi optimal. Pemilihan lawan yang membutuhkan sedikit energi untuk memenangkan pertempuran merupakan bagian dari strategi menyerang. Untuk mengoptimalkan pemilihan lawan tersebut diperlukan suatu metode pengambilan keputusan. Bagaimana mendapatkan pilihan urutan lawan yang tepat supaya strategi menjadi optimal merupakan hal yang perlu dipertimbangkan [16]. Dalam penelitian ini menggunakan strategi menyerang karena pahlawan dalam *game* ini berperang penting dalam sebuah peperangan.

2.3.2 Strategi Bertahan

Bertahan merupakan bagian dari strategi dimana gaya bertahan bergantung kepada medan pertempuran jumlah anggota, kekuatan lawan, dan senjata untuk melindungi. Pada pertempuran, senjata pelindung seperti baju atau rompi serta tameng juga mempengaruhi gaya bertahan suatu tim. Koordinasi dan kekompakan tim dalam melindungi teman yang lemah juga menjadi kunci dari keberhasilan strategi bertahan. Tidak hanya digunakan untuk tujuan menghemat amunisi atau kekuatan, tetapi dapat juga digunakan untuk menguras tenaga dan amunisi lawan sebelum mulai menyerang.

2.3.3 Strategi Menghindar

Menghindar juga bagian dari strategi dimana gaya menghindar bisa bervariasi, misal : menghindar dengan bersembunyi atau menghindar dengan memutar rute yang dilalui. Strategi menghindar ini juga bisa digunakan pada saat lawan yang dihadapi mempunyai jarak serang yang pendek sehingga saat diserang bisa mundur atau berpindah posisi. Penghematan kekuatan atau amunisi juga bisa dengan menggunakan strategi ini. Strategi ini juga bermanfaat untuk anggota kelompok yang hanya mempunyai tugas sebagai mata – mata atau *scout* dalam pertempuran untuk mencari informasi tentang musuh yang dihadapi.

2.4 Operasi Penyerangan

Seorang pemain melakukan operasi penyerangan dengan menghancurkan instalasi lawan dan meminimalkan kerusakan pada instalasi dan aset pemain sendiri. Sebuah serangan pemain mengalokasikan sebuah pengumpulan aset penyerangan pada platform terbang untuk satu set target lawan dan ancaman di tanah. Masalahnya Dinamis : Cuaca dan faktor lingkungan lain yang mempengaruhi kinerja aset, ancaman yang tidak diketahui dapat dikirimkan dan menjadi target yang dihancurkan [5].

Menyerang memerlukan strategi yang tepat agar serangan yang dilakukan untuk mengalahkan tim lawan menjadi optimal. Tingkat ke-Optimal-an dari serangan dapat diukur dari banyaknya atau sedikitnya rata – rata health dari tim di akhir pertarungan. Ke-Optimal-an dari strategi menyerang dapat diukur dari banyak sedikitnya korban yang tewas & luka – luka atau diukur dari jumlah health yang tersisa dari setiap anggota kelompok [4].

Bagaimanapun model strategi menyerang, umumnya mempunyai tujuan akhir mengalahkan musuh. [17]

2.5 Optimasi

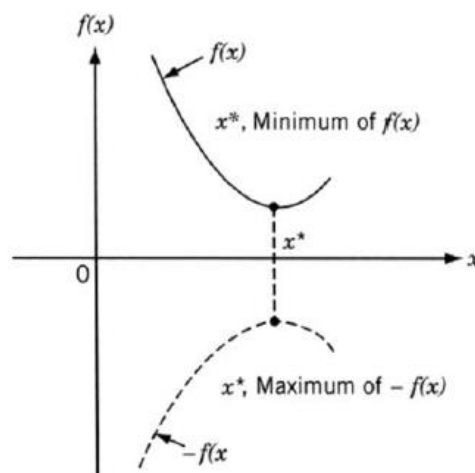
Teknik optimasi merupakan suatu cara yang dilakukan untuk memberikan hasil terbaik yang diinginkan. Teknik optimasi ini banyak memberikan manfaat dalam mengambil keputusan. Dasar dari pengambilan keputusan ini antara lain :

1. Memilih alternatif yang terbaik
2. Perumusan berbagai alternatif tindakan sesuai dengan yang sedang dalam perhatian, dan dalam pemilihan alternatif yang tepat
3. Keefektifan dalam mencapai tujuan yang dikehendaki dalam mengambil keputusan.

Optimasi juga merupakan sebuah proses produksi yang lebih efisien (lebih kecil dan / atau lebih cepat) yang mana programnya melalui seleksi dan desain struktur data, algoritma, dan urutan instruksi. Dalam segi bahasa, Optimal = paling bagus/tinggi; tertinggi; terbagus; paling menguntungkan. Optimal (ter) baik: tertinggi; paling menguntungkan. Ray Fernand mengatakan bahwa optimasi sangat

berguna di hamper segala bidang dalam rangka melakukan usaha secara efektif dan efisien untuk mencapai target hasil yang ingin dicapai [18].

Optimalisasi adalah tindakan untuk memperoleh hasil yang terbaik dengan keadaan yang diberikan. Dalam desain konstruksi, dan pemeliharaan dari system teknik, harus diambil beberapa teknologi dan keputusan managerial dalam beberapa tahap. Tujuan akhir dari semua keputusan seperti ini adalah meminimalkan upaya yang diperlukan atau untuk memaksimalkan manfaat yang diinginkan. Mengacu pada pendapat Singiresu S Rao, John Wiley dan Sons (2009) optimalisasi juga dapat didefinisikan sebagai proses untuk mendapatkan keadaan yang memberikan nilai maksimum atau minimum dari suatu fungsi. Hal ini dapat dilihat di Gambar 2.15, bahwa jika titik x^* berkaitan dengan nilai minimum fungsi $f(x)$, titik yang sama juga berkaitan dengan nilai maksimum dari negative fungsi tersebut $-f(x)$. Tanpa menghilangkan keumumannya, optimasi dapat diartikan meminimalkan, karena maksimu suatu fungsi dapat diperoleh melalui minimum dari negatif fungsi yang sama. Optimasi digunakan pada penelitian ini untuk perilaku pahlawan yang *Multiobjective Optimization*. Penjelasan tentang *Multiobjective Optimization* akan dijelaskan pada bab selanjutnya.



Gambar 2.4 : Minimum dari $f(x)$ sama dengan Maksimum dari $-f(x)$

2.5.1 Teknik Optimasi

Optimasi digunakan untuk menterjemahkan kata optimization. Sebenarnya lebih tepat menggunakan kata optimasi, tetapi karena istilah optimasi sudah sering digunakan, maka digunakanlah kata optimasi sebagai terjemahan optimization

[19]. Optimasi sudah berkembang sejak newton di abad 17 menemukan cara menghitung akar. Saat inipun ilmu optimasi masih berkembang dalam hal teknik maupun aplikasi. Banyak kasus dalam kehidupan sehari – hari yang melibatkan optimasi untuk memecahkannya. Algoritma optimasi terbagi menjadi 2 jenis, yaitu Algoritma optimasi dengan pendekatan berbasis deterministic dan Algoritma optimasi dengan pendekatan berbasis probabilistic. Yang termasuk dalam algoritma berbasis deterministic diantaranya *State Space Search*, *Dynamic Programming*, dan *Branch and Bound*. Sedangkan algoritma optimasi yang termasuk dalam algoritma dengan probabilistic diantaranya algoritma Monte Carlo dengan berbagai keturunannya, *Ant Colony Optimization* (ACO), dan *Particle Swarm Optimization* (PSO) [20].

2.5.2 Metode Optimasi

Metode mencari optimum dikenal sebagai teknik *mathematical programming* dan bisa dipelajari sebagai bagian riset operasi. Riset operasi adalah cabang matematika yang berkaitan dengan penerapan metode ilmiah dan teknik pengambilan keputusan dan penetapan penyelesaian terbaik atau optimal. Pada awal dari subyek riset operasi dapat ditelusuri pada periode awal Perang Dunia II, selama perang, militer inggris menghadapi masalah mengalokasikan sumber daya yang sangat langka dan terbatas (seperti pesawat tempur, radar, dan kapal selam) untuk beberapa kegiatan (penyebaran ke berbagai target dan tujuan). Karena tidak ada metode sistematis yang tersedia untuk memecahkan masalah alokasi sumber daya, militer diatas (tim matematikawan) mengembangkan metode untuk memecahkan masalah secara ilmiah. Metode yang dikembangkan oleh tim berperan penting dalam memenangkan pertempuran udara oleh inggris. Metode tersebut seperti program linier, yang dikembangkan sebagai hasil riset pada militer.

Perkembangan metode optimasi semakin mengalami kemajuan hingga masa modern, hal ini dapat dilihat dengan semakin banyak metode optimasi yang ditemukan dan dapat menghasilkan solusi yang semakin optimal. Metode optimasi yang populer dan banyak dipakai antara lain seperti *Dynamic Programming*, *Integer Programming*, *Game Theory*, dan metode optimasi modern. Metode optimasi modern juga disebut metode optimasi non-tradisional, muncul sebagai

metode yang ampuh dan populer untuk menyelesaikan masalah teknik optimasi yang kompleks. Metode yang termasuk seperti *algoritma genetic*, optimasi partikel *swarm*, optimasi koloni semut, optimasi berbasis jaringan syaraf tiruan, optimasi fuzzy, dan *simulated annealing*.

2.6 Perilaku

Perilaku dapat didefinisikan sebagai cara di mana seorang individu berperilaku atau bertindak. Ini adalah cara seorang individu melakukan sendiri / dirinya sendiri. Perilaku harus dilihat dalam referensi untuk sebuah fenomena, suatu benda atau orang. Hal ini dapat dilihat dalam referensi untuk norma-norma masyarakat, atau cara di mana seseorang memperlakukan orang lain atau menangani benda. Perilaku, oleh karena itu, adalah cara seseorang bertindak terhadap orang-orang, masyarakat atau benda. Hal ini dapat baik buruk atau baik. Hal ini dapat normal atau abnormal sesuai dengan norma-norma masyarakat. Masyarakat akan selalu mencoba untuk memperbaiki perilaku buruk dan mencoba untuk membawa perilaku abnormal kembali normal [21].

2.7 Pahlawan

Definisi pahlawan akan dijelaskan menurut 2 versi kamus besar dunia, yaitu Kamus Merriam Webster (Kamus Amerika) dan Kamus Oxford (Kamus Inggris). Berikut Definisi *Hero* (Pahlaman) menurut mereka :

1. Dalam Kamus Merriam Webster (Kamus Amerika)

- a. Orang yang dikagumi karena tindakan besar atau berani atau kualitas baik
- b. Seseorang yang sangat dikagumi
- c. Karakter laki-laki utama dalam cerita, drama, film, dll

Definisi Penuh *Hero* :

- a. Tokoh mitologi atau legenda sering dikatakan keturunan ilahi diberkahi dengan kekuatan besar atau kemampuan
- b. Sebuah prajurit terkenal
- c. Seorang pria dikagumi karena prestasinya dan kualitas mulia
- d. Yang menunjukkan keberanian besar

- e. Karakter utama laki-laki dalam sebuah karya sastra atau dramatis
- f. Tokoh sentral dalam sebuah acara, periode, atau gerakan

2. Dalam Kamus Oxford (Kamus Inggris)

- a. Seseorang, biasanya seorang pria, yang dikagumi karena keberaniannya , prestasinya yang luar biasa, atau kualitasnya yang mulia : contoh = pahlawan perang.
- b. Karakter utama pria dalam sebuah buku, permainan, atau film, yang biasanya diidentifikasi dengan kualitas yang baik, dan diharapkan pembacanya bersimpati (Dalam mitologi dan cerita rakyat) orang dari kualitas super dan sering berasal dari setengah Tuhan, dalam satu tertentu yang di eksploitasi adalah subjek dari mitos Yunani kuno.

2.8 Perilaku Pahlawan

Pahlawan pasti memiliki jiwa dan sikap kepemimpinan. Dari tahun 1950 sampai tahun 1970-an, cara menonjol untuk konsep perilaku kepemimpinan adalah dalam hal kepedulian terhadap tujuan tugas dan terhadap kepedulian bagi orang-orang (atau hubungan). Studi awal Kepemimpinan Negara Ohio (Fleishman , 1953) memberikan bukti tentang hubungan tugas dan perilaku. Studi tentang implikasi dari dua perilaku untuk efektivitas kepemimpinan belum membuahkan hasil yang konsisten (Yukl , 1998). Survei Penelitian menggunakan kuesioner deskripsi perilaku gagal memberikan banyak dukungan untuk gagasan bahwa pemimpin yang efektif memiliki skor yang tinggi pada kedua dimensi . Sebuah versi halus dari teori tinggi adalah bahwa pemimpin yang efektif mengintegrasikan tugas dan kekhawatiran orang-orang dengan cara yang relevan untuk situasi, daripada hanya menggunakan tugas dan perilaku hubungan (Blake & Mouton , 1982 ; Sashkin & Fulmer , 1988) [9]. Dalam penelitian ini pahlawan dalam *game* DWIPA YUDHA memiliki beberapa perilaku yang mencerminkan seorang pahlawan pada umumnya, pahlawan yang kita ketahui di dunia nyata maupun dalam film. Pahlawan dalam penelitian memiliki 5 perilaku yang akan dioptimasi yaitu : *Agresif* (penyerang), *Supportif* (pendukung atau pelindung), *Leadership* (pemimpin), *Egois* 1 (melarikan

diri), dan Egois 2 (mengajak pasukannya juga melarikan diri). Penjelasan setiap perilaku adalah sebagai berikut :

2.9 *Big Five Personality* (Lima Besar Kepribadian)

Pahlawan dalam penelitian ini memiliki 5 perilaku yang merupakan sifat dasarnya. Pahlawan dalam penelitian ini memiliki 5 perilaku berbeda yang saling bertentangan yang terdiri dari *Agresif*, *Supportif*, *Leadership*, Egois 1, dan Egois 2. Kelima perilaku tersebut akan dioptimasi dan untuk melihat hasilnya, maka digunakanlah diagram *big five personality* yang juga sering kita jumpai pada *game PES (Pre Evolution Soccer)*. Banyak psikolog kepribadian kontemporer percaya bahwa ada lima dimensi dasar kepribadian, sering disebut sebagai kepribadian "*Big 5*". Lima ciri-ciri kepribadian yang luas dijelaskan oleh teori yang *extraversion*, *agreeableness*, *openness*, *conscientiousness*, dan *neuroticisme*. Teori sifat Sebelumnya menyarankan berbagai jumlah kemungkinan ciri-ciri , termasuk Gordon Allport daftar 4.000 karakter kepribadian, Raymond Cattell 's 16 faktor kepribadian , dan teori tiga faktor Hans Eysenck ini. Namun, banyak peneliti merasa bahwa teori Cattell itu terlalu rumit dan Eysenck ini terlalu terbatas dalam lingkup. Akibatnya, teori lima faktor muncul untuk menggambarkan sifat-sifat dasar yang berfungsi sebagai blok bangunan kepribadian. Apakah Lima Besar Dimensi Kepribadian?

Lima Besar Dimensi Kepribadian atau *Big Five Personality* merupakan pendekatan dalam psikologi kepribadian yang mengelompokan sifat kepribadian dengan analisis faktor. Tokoh pelopornya adalah Allport dan Cattell. *Big Five Personality* adalah suatu pendekatan yang digunakan dalam psikologi untuk melihat kepribadian manusia melalui sifat yang tersusun dalam lima buah domain kepribadian yang telah dibentuk dengan menggunakan analisis faktor. Lima sifat kepribadian tersebut adalah *extraversion*, *agreeableness*, *conscientiousness*, *neuroticism*, *openness* pada pengalaman.

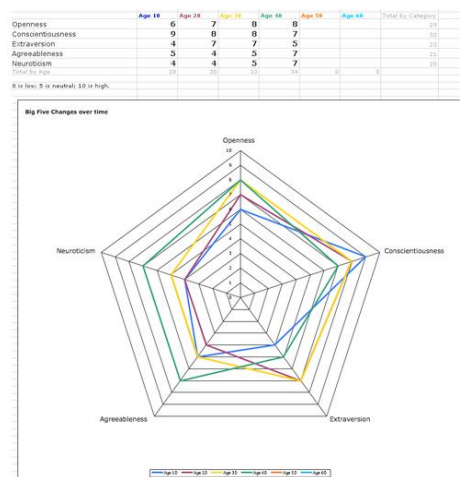
2.8.1 Penelitian *Big Five Personality*

McCrae dan rekan-rekannya [22] juga menemukan bahwa lima besar sifat juga sangat universal. Satu studi yang tampak pada orang-orang dari lebih dari 50

budaya yang berbeda menemukan bahwa lima dimensi dapat secara akurat digunakan untuk menggambarkan kepribadian. Berdasarkan penelitian ini, banyak psikolog sekarang percaya bahwa lima dimensi kepribadian tidak hanya universal; mereka juga memiliki asal-usul biologis. Psikologi David Buss [23] telah mengusulkan bahwa penjelasan evolusi selama lima ciri-ciri kepribadian inti, menunjukkan bahwa ciri-ciri kepribadian ini mewakili kualitas paling penting yang membentuk lanskap sosial kita.

2.8.2 Pikiran Akhir

Selalu ingat perilaku yang melibatkan interaksi antara kepribadian yang mendasari seseorang dan variabel situasional. Situasi bahwa seseorang menemukan dirinya sendiri dalam memainkan peran utama dalam bagaimana orang bereaksi. Namun, dalam banyak kasus, orang menawarkan respon yang konsisten dengan sifat-sifat kepribadian yang mendasarinya. Dimensi ini mewakili daerah yang luas dari kepribadian. Penelitian telah menunjukkan bahwa pengelompokan ini karakteristik cenderung terjadi bersama-sama dalam banyak orang. Misalnya, individu yang bergaul cenderung banyak bicara. Namun, sifat-sifat ini tidak selalu terjadi bersama-sama. Kepribadian adalah orang yang kompleks dan bervariasi dan masing-masing akan menampilkan perilaku di beberapa dimensi ini. Diasumsi dari *Big Five Personality*, pembuatan grafik lima perilaku optimal pahlawan akan dijelaskan lebih lanjut pada bab setelah ini. Dan berikut gambar grafik *Big Five Personality* :



Gambar 2.5 : Grafik *Big Five Personality*

2.10 Agresif (Penyerang)

Agresif dalam hal komunikasi mempengaruhi respon balik orang lain terhadap kita. Bahkan jika kita ingin bertindak tegas, kita mungkin tidak bisa jika kita tidak mengenali perbedaan gaya komunikasi. Misalnya, perilaku agresif kadang-kadang salah diidentifikasi sebagai tegas. Itu adalah penting bagi kita untuk dapat benar mengidentifikasi gaya kita sendiri dan orang lain dari komunikasi. Harus ada pesan yang konsisten dikirim dalam komunikasi verbal dan non-verbal Anda [24].

2.9.1 Karakteristik Komunikasi Agresif

A. Definisi:

1. Anda berdiri untuk hak-hak pribadi dan mengekspresikan pikiran, perasaan dan keyakinan dengan cara yang biasanya tidak tepat dan selalu melanggar hak-hak orang lain.
2. Orang sering merasa hancur oleh perjumpaan dengan orang yang agresif.
3. Keunggulan dipertahankan dengan menjatuhkan orang lain.
4. Ketika terancam Anda menyerang.

B. Karakteristik Verbal:

1. Melengking, sarkastik atau merendahkan suara
2. Fasih, beberapa keragu-raguan
3. Seringkali tiba-tiba dipotong
4. Sering cepat
5. Menekankan kata menyalahkan
6. Suara tegas
7. Nada sarkastis, dingin, keras
8. Suara bisa melengking, sering berteriak, naik di akhir
9. Penggunaan ancaman, misalnya, "Anda sebaiknya berhati-hati" atau "Jika Anda tidak ..."
10. Merendah, misalnya "Kau pasti bercanda ..." atau "Jangan begitu bodoh"
11. Komentar evaluatif, menekankan konsep-konsep seperti: harus ", "buruk ", "seharusnya "

12. Seksual / komentar rasis
13. Kesombongan, misalnya, "Saya tidak punya masalah seperti Anda"
14. Pendapat yang dikemukakan sebagai fakta, misalnya, "Tak seorang pun ingin berperilaku seperti itu" atau "Itu cara yang berguna untuk melakukannya"
15. Mengancam pertanyaan, misalnya, "Apakah kau tidak selesai belum?" Atau "Kenapa kau melakukannya seperti itu?"

C. Karakteristik non-verbal:

1. Mengganggu ke ruang orang lain
2. Melotot ke orang lain
3. Gerakan seperti menunjuk, mengepalkan tinju
4. Mondar-mandir tak sabar
5. Condong ke depan atau lebih
6. Crossing lengan (didekati)
7. Tersenyum bisa menjadi mencibir
8. Cemberut saat marah
9. Set rahang yang kuat

D. Gaya Berpikir :

1. "Aku akan menguasai Anda sebelum Anda memiliki kesempatan untuk mendapatkan saya"
2. "Aku keluar untuk nomor satu"
3. "Dunia adalah tanah pertempuran dan saya keluar untuk menang"

E. Hadiah atau Imbalan:

1. Anda membuat orang lain melakukan perintah Anda
2. Hal yang cenderung pergi di jalanmu
3. Anda lebih sedikit rapuh
4. Anda menyukai perasaan yang berada dalam kendali
5. Pelepasan ketegangan
6. Anda merasa kuat

F. Harga:

1. Perilaku Anda akan membuat musuh dan kebencian pada orang di sekitar Anda
2. Hal ini dapat mengakibatkan rasa paranoia dan ketakutan

Informasi diambil dari *The Centre For Clinical Intervention's series di Assertiveness

Mungkin sebagian besar orang, termasuk psikolog, akan setuju secara umum dengan definisi yang diberikan oleh Buss (1961: 1): agresi adalah 'respon yang memberikan rangsangan berbahaya untuk organisme lain'. Secara pasti apa yang biasanya kita sebut agresi melibatkan rangsangan permusuhan dari sejenisnya dan intensitasnya, baik itu dalam bentuk peluru, bom fragmen, pukulan fisik, atau tindakan yang lebih halus seperti penghinaan atau kritik tidak layak. Kami tidak menemukan masalah nyata yang menyatakan apa itu agresi, sampai kita mencoba merumuskan definisi yang berlangsung melampaui perilaku membahayakan yang sederhana. Agresi tidak sesederhana sebagai murni definisi perilaku yang akan ditunjukkan. Unsur-unsur lain harus ditambahkan, dan elemen-elemen ini membuat kompleksitas tertentu. Perilaku *agresif* seorang pahlawan dalam penelitian ini lebih mengarah pada sifat pahlawan yang kita ketahui dalam dunia nyata maupun dalam film bahwa ke-*agresif*-annya dalam hal menyerang musuhnya. Pahlawan rela mati demi membela sebuah kebenaran. Bila dalam *game* RTS, membela kebenaran adalah membela negaranya atau membela desanya agar tetap aman, membela dari musuh yang ingin menghancurkan Negara atau desanya tersebut. Russell G. Geen (2001) dalam *Open University Press* buku edisi keduanya menjabarkan lebih luas tentang Agresi. Berikut bagian dari bukunya yang menjelaskan lebih rinci :

2.9.2 Definisi Kerja Agresi [25]

Dengan semua pikiran sebelumnya, kita sekarang dapat menyarankan definisi kerja dari agresi manusia: agresi adalah pengiriman stimulus yang tidak menyenangkan dari satu orang ke orang lain, dengan maksud untuk menyakiti dan

dengan harapan menyebabkan kerugian tersebut, ketika orang lain termotivasi untuk melarikan diri atau menghindari stimulus. Hal ini mengakui bahwa definisi ini tidak mencakup semua contoh dan bahwa hal itu dapat diserang di beberapa titik. Meskipun juga tidak dimulai berurusan dengan banyak variabel yang terlibat dalam agresi. Seperti misalnya, tidak menyebutkan peran yang dimainkan oleh emosi dalam banyak tindakan agresif. Ini tidak memperhitungkan penghakiman kognitif yang kompleks yang sering mendahului agresi. Tidak memperhitungkan fakta bahwa agresi sering timbal balik, atau bahwa kadang-kadang berfungsi sebagai 'Katup pengaman' berharga untuk seseorang yang mengalami tingkat kemarahan ekstrim. Dan banyak masalah lain seperti melanggar semua atas definisi dasar agresi, namun mereka melampaui yang sederhana dan bahkan terdiri dari banyak variabel yang merupakan sebagian besar materi dari buku ini.

2.9.3 Bentuk Agresi

Ketika mereka mendengar kata 'agresi' kebanyakan orang mungkin pertama cenderung untuk memikirkan kekuatan fisik - tinju-pertarungan, serangan dengan senjata, keras retort lisan atau bentuk lain dari tindakan intens dan hukuman yang berlaku dalam perjalanan konflik antara dua orang. Sebenarnya, menurut Definisi yang telah kami adopsi, agresi dapat dilakukan di perilaku apapun yang digerakkan oleh niat untuk menyakiti orang lain oleh keinginan seseorang. Gosip jahat tentang seseorang dengan harapan bahwa merusak reputasi seseorang akan dianggap agresi. Demikian juga, merusak atau menghancurkan properti orang lain bisa menjadi cara yang sangat efektif untuk aggressing terhadap orang itu. Bahkan sesuatu yang halus dan terkendali sebagai penghinaan social yang bisa menjadi sumber kuat untuk merugikan korban, sebuah bahaya yang jelas dimaksudkan oleh orang memberikan itu.

2.9.4 Agresi Proaktif dan Reaktif

Sejumlah studi terbaru agresi menarik perbedaan antara agresi reaktif dan proaktif (misalnya Crick dan Dodge 1996). Yang pertama istilah-istilah ini mengacu pada perilaku agresif yang diberlakukan dalam menanggapi provokasi, seperti serangan atau penghinaan, dan diwujudkan dalam baik pertahanan diri dan

tindakan marah. Istilah yang terakhir mengacu pada agresi yang dimulai tanpa provokasi jelas, seperti yang kita lihat di perilaku *bullying*. Perilaku tersebut tidak ditimbulkan oleh kemarahan, permusuhan atau kebutuhan untuk membela diri, tetapi dengan motif lain yang berhubungan dengan memperoleh barang, tenaga penegasan, menjamin referensi persetujuan kelompok dan tujuan seperti lainnya. Agresi reaktif dan proaktif adalah setara dengan apa yang disebut oleh ahli teori sebelumnya yaitu agresi afektif dan instrumental. Afektif-instrumental (atau reaktif-proaktif) diferensiasi akan muncul dalam konteks tertentu dalam buku ini, dan pembaca hendaknya memikirkan perbedaan antara kedua jenis agresi.

Pengingat dari buku ini dikhususkan terutama untuk diskusi tentang proses yang terlibat dalam agresi afektif. Penekanan ini sama sekali tidak dimaksudkan untuk menyiratkan bahwa agresi instrumental itu penting. Namun, belum diteliti hampir memiliki kedalaman yang sama seperti yang dimiliki agresi afektif. Ada, misalnya, tidak ada sebagian besar dari literatur yang menganalisis variabel terlibat ketika seseorang menyakiti yang lain demi uang, juga tidak memiliki data luas dari studi terkontrol pada mediator untuk membela diri. Beberapa penelitian telah mulai menyelidiki antededen dari setiap jenis agresi, dan mereka akan dikutip. Di sisi lain, kita memiliki besar Kesepakatan informasi tentang agresi afektif dan proses yang kontribusinya untuk itu.

2.9.5 Pembelajaran sosial

Antededen agresi yang telah kita bahas sejauh ini dalam hal ini Bagian semuanya telah didasarkan pada sistem biologis manusia. Pada suatu waktu, selama kekuasaan dari behaviorisme, variabel tersebut belum banyak ditetapkan di agresi; Perilaku tersebut akan dijelaskan hampir seluruhnya sebagai sesuatu yang diperoleh melalui pengondisian dan pembelajaran. Hari ini, tentu saja, kami menyadari bahwa peran dari belajar dan faktor bawaan dalam agresi manusia tidak dapat dijelaskan dalam baik / atau cara. Hampir setiap psikolog yang menyelidiki masalah mengakui bahwa keduanya terlibat dan bahwa perbedaan sudut pandang melibatkan penekanan relatif pada setiap penempatan. Untuk mengatur 'alami' terhadap 'pemeliharaan' di pembahasan agresi manusia adalah untuk menciptakan dikotomi

palsu. Di tempat lain di Buku ini disarankan bahwa baik belajar dan keturunan yang terbaik dipahami berdiri sebagai variabel latar belakang yang membuat tingkat potensi agresi tanpa nenek moyangnya langsung. Perilaku agresif adalah respon kondisi dalam situasi yang memprovokasi orang; bahkan ketika salah satu dijual untuk agresi dan mampu berperilaku agresif, sebuah situasi tertentu harus menimbulkan tindakan. Probabilitas bahwa perilaku seperti itu akan terjadi, dan juga intensitas perilaku, akan bervariasi sesuai dengan baik dari sifat provokasi dan tingkat potensi agresi yang ditetapkan oleh beberapa variabel latar belakang. Tentu orang yang lahir dengan disposisi menjadi kekerasan akan lebih agresif ketika diserang daripada yang kurang disposisi tersebut, dan orang-orang yang telah memperoleh kecenderungan agresif yang kuat melalui pembelajaran sosial akan bereaksi lebih agresif daripada mereka yang belum. Keturunan dan pembelajaran sosial merupakan faktor pelengkap dalam agresi manusia.

Teori pembelajaran sosial agresi muncul pada 1960-an, sebagian besar sebagai akibat dari teori Albert Bandura dan rekan-rekannya. Pendekatan itu telah mengalami beberapa elaborasi sejak pertama kali disajikan dan terus mengarahkan pengaruh yang kuat. Ini menekankan akuisisi dan pemeliharaan kecenderungan respon agresif. Meskipun tidak mengesampingkan provokasi sebagai kontributor penting untuk agresi, pendekatan sosial belajar memperlakukan acara seperti kondisi di mana belajar perilaku agresif dapat diberlakukan. Demikian juga, teori termasuk pengakuan faktor biologis dalam agresi tanpa mengenai faktor tersebut sebagai penyebab langsung dari perilaku agresif. Sebaliknya, teori mengasumsikan genetic yang abadi dan biologis seseorang menciptakan potensi agresi, sedangkan spesifik perilaku agresif bentuk dan frekuensi, situasi yang membangkitkan dan target ke arah yang itu diarahkan - diperoleh melalui pengalaman (Bandura 1983). Pembelajaran sosial terdiri dari akuisisi tanggapan melalui observasi dan pemeliharaan perilaku melalui penguatan. Normalisasi tersebut anak mal mengamati banyak contoh agresi baik dalam kehidupan nyata situasi di rumah, di sekolah dan di jalan-jalan, dan fantasi dunia televisi dan film. Dengan mengamati konsekuensi agresi untuk aktor, anak secara bertahap mengakuisisi dasar pengetahuan tentang aturan-aturan tertentu perilaku (misalnya satu yang kadang-kadang mungkin mendapatkan sesuatu yang diinginkan dengan menggunakan

kekerasan). Dengan cara ini repertoar/khasanah perilaku agresif dibangun. Apakah perilaku tersebut bertindak keluar tergantung pada kontinjensi bahwa anak merasakan untuk perilakunya.

Jika insentif cocok untuk agresi yang hadir, kemungkinan agresi cenderung tinggi. Selain itu, anak mengalami penghargaan dan konsekuensi menghukum berdasarkan agresi; perilaku agresi diperoleh melalui pengamatan kemungkinan akan dilakukan hanya jika anak telah dihargai untuk tindakan tersebut. Bandura (1986) menyatakan bahwa proses pembelajaran sosial bergantung tentang pembentukan representasi mental anak di peristiwa sosial lingkungan hidup. Imbalan dan hukuman untuk agresi direpresentasikan dalam bentuk harapan dari hasil masa depan agresi dan utilitas atau nilai bahwa agresi dimiliki individu. Misalnya, seorang anak dengan sejarah mencapai tujuan yang berharga melalui intimidasi anak-anak lain di sekolah segera datang ke percayaan bahwa *bullying* lebih lanjut dalam pengaturan yang sama akan terus memberikan imbalan berharga. Akibatnya, agresi mengakuisisi tingkat tinggi nilai dan kemungkinan akan diulang pada saat yang tepat pada isyarat situasional yang hadir.

Selanjutnya, selain membentuk diharapkan dari kemungkinan bahwa agresi akan dihargai, yang disebut hasil harapan, anak diperkuat juga mengembangkan rasa percaya diri dalam kemampuannya untuk diperlukan menjalankan perilaku agresif, disebut keberhasilan harapan diri sendiri. Bukti untuk kedua jenis harapan itu kembali di porting oleh Perry et al . (1986), yang menemukan bahwa anak-anak digambarkan sebagai sangat agresif dengan rekan-rekan mereka mengungkapkan keyakinan yang lebih besar dalam kemampuan mereka untuk melaksanakan solusi agresif untuk konflik antar pribadi dari kurang agresi anak komprehensif, serta keyakinan yang lebih besar bahwa agresi akan menghasilkan imbalan nyata dan akan berhasil dalam forestalling konflik di masa depan. Dalam sebuah studi dari komponen nilai agresivitas kebiasaan, Boldizar et al. (1989) menemukan bahwa sesama nilai tinggi agresivitas diprediksi positif datang pada anak-anak yang berhubungan dengan hasil agresi. Anak Anak dinilai sebagai agresivitas tinggi terpasang nilai positif yang lebih besar daripada anak-anak kurang agresif untuk 'mengendalikan korban' yang dihasilkan dari agresi terhadap nilai terakhir, dan kurang negatif pada hasil seperti korban penderitaan, ancaman pembalasan,

penolakan oleh rekan-rekan dan perasaan negatif tentang diri mereka sendiri. Singkatnya, anak-anak yang sangat agresif melihat lebih hasil yang baik timbul dari agresi, dan yang buruk lebih sedikit, daripada kurang anak yang agresif.

2.11 *Supportif* (Pendukung atau Pelindung) [26]

Perilaku *supportif* dalam penelitian ini merupakan perilaku pahlawan yang dapat melindungi dan mendukung pasukannya. Pahlawan akan melindungi dan mendukung pasukannya yang lemah, yang lebih rendah daripadanya. Yang lebih rendah kesehatannya dan mendapatkan *damage* lebih besar sehingga mengancam nyawanya. *Supportive Behaviour* (Perilaku Mendukung) adalah Perilaku pemimpin yang memberikan pujian, mendengarkan, memberikan semangat, melibatkan anggota tim dalam mengambil keputusan. Pada perilaku *Supportive*, Pemimpin dan anggota tim membuat keputusan bersama. Peran pemimpin adalah untuk menampung fasilitas, mendengarkan, memberikan umpan, semangat dan dukungan. Dalam memberikan dukungan, pemimpin memberikan dorongan untuk dilaksanakannya komunikasi ke atas, mencari ide – ide, mendengarkan, memberikan nasihat, memberikan pengakuan, dan membantu orang – orang yang dipimpin dalam membuat dan mengambil keputusan. Gaya ini akan membantu mengembangkan orang jika digunakan secara efektif. Pemimpin membimbing anggota untuk membuat keputusan yang baik. Selain itu, pemimpin menekankan bahwa anggota tim harus mengambil keputusan dengan bimbingannya. Sisi negatif gaya ini adalah kecenderungan untuk terlalu akomodatif.

2.12 *Leadership* (Kepemimpinan) [26]

Kepemimpinan dimulai dari hati bukan kepala. Gambaran kepemimpinan telah berubah secara dramatis, pada saat ini. Memang, kepemimpinan mengalami perkembangan sesuai dengan zaman dan masa yang terus mengalami perubahan. Ilmu – ilmu kepemimpinan dulu, sudah tidak relevan lagi untuk diterapkan pada masa kini. Kepemimpinan yang secara teoretis dan akademis pun kadang tidak semenarik dibanding dengan kepemimpinan praktis. Namun, ada nilai – nilai lama yang tidak akan usung, seperti tanggung jawab, integritas, komitmen, karakter, dan ciri – ciri yang mencerminkan sikap mentap positif. Nilai – nilai ini mungkin kuno,

tetapi tidak akan using oleh waktu, serta akan tetap ada, dan sudah teruji melalui waktu sampai kini.

Dalam menyikapi fenomena tersebut, dibutuhkan suatu pribadi yang mempunyai karakter yang berkualitas, seperti “tahan banting”, kuat, dinamis, dan kepemilikan kepercayaan pribadi atas kemampuan dan melakukan tindakan yang penuh asih, tanpa mengalahkan dan merendahkan orang lain. Segala frustrasi, ketakutan, buang waktu percuma, dan hal – hal yang menjadikan tidak efektif, akan membuat kita jauh dari pemenang.

Leadership atau kepemimpinan adalah proses seorang individu mempengaruhi anggota – anggota kelompok lainnya untuk pencapaian tujuan kelompok atau organisasi. *Leadership* adalah suatu proses yang melibatkan proses mempengaruhi, yakni suatu proses seorang pemimpin mengubah tindakan atau perilaku beberapa anggota kelompok atau bawahan. Secara umum, *leadership* berkaitan dengan penggunaan teknik memengaruhi yang tidak memaksa. Hal ini berarti bahwa *leadership* mendasarkan diri pada perasaan positif antara pemimpin dan yang dipimpin. Dengan kata lain, bawahan menerima pengaruh dari pemimpin, karena pemimpinnya mempunyai kepribadian dan kemampuan untuk memimpin sehingga disukai dan dihormati, bukan hanya karena para pemimpin tersebut mempunyai posisi memegang jabatan dari kekuasaan secara formal.

Leadership melibatkan penggunaan pengaruh untuk satu maksud tertentu, yakni untuk mencapai tujuan kelompok atau tujuan organisasi. Dengan kata lain, para pemimpin memfokuskan diri pada pengarahan tindakan atau perilaku para bawahan mereka untuk tercapainya tujuan spesifik; para *leader* tidak menaruh perhatian terhadap pengarahan tindakan atau perilaku yang tidak relevan dengan pencapaian tujuan organisasi atau kelompok. *Leadership* merupakan satu proses dua arah. Pemimpin sudah barang tentu memengaruhi bawahan dengan berbagai cara, namun sebaliknya, para pemimpin sering kali dipengaruhi oleh bawahan. Pemimpin harus memiliki integritas. Integritas adalah suatu prinsip yang didasarkan atas karakter, etika, agama, dan moral yang baik.

Menjadi pemimpin yang sukses adalah perjalanan seumur hidup. Yang pertama harus dilakukan untuk menjadi pemimpin sukses adalah mempunyai motivasi, dan menyatakan niat dengan bebas. Segala tujuan tanpa motivasi tidak

akan tercapai. Tanpa motivasi, sulit melakukan apa saja dengan baik. Dengan motivasi segala sesuatunya menjadi mungkin. Pemimpin yang mempunyai motivasi tinggi adalah pemimpin yang bisa mencetak hasil – hasil yang mengejutkan. Pemimpin inilah yang menawarkan kepada kita harapan dibangunnya sebuah dunia organisasi atau bisnis yang berdasarkan kebenaran dan pertanggung jawaban. Sukses kepemimpinan adalah kepemimpinan yang menciptakan langkah paling menarik dari organisasi untuk jangka panjang.

2.11.1 Kepemimpinan dan Karakteristik Pribadi

Kepemimpinan bukan sesuatu yang secara alami diturunkan dalam arti secara biologis. Apabila seorang pemimpin, anaknya belum tentu akan menjadi pemimpin. Ratu Elizabeth I dan Abraham Lincoln tampak berbeda dibanding manusia biasa dalam beberapa hal, mereka tampak memiliki ambisi tinggi dengan visi yang jelas. Para politisi, pahlawan olahraga, *top executive* memiliki suatu aura yang menempatkan mereka tampak berbeda dengan orang kebanyakan.

Karakter membedakan para pemimpin besar dengan kebanyakan orang. Pemimpin mempunyai karakter yang stabil sepanjang waktu, dan dalam lingkungan yang berbeda. Oleh karena itu, bisa dikatakan bahwa karakter yang dipunyai para pemimpin besar tidak memandang kapan dan di mana mereka hidup. Orang yang mempunyai kemampuan penguasaan diri yang tinggi, dan mempunyai suatu kebutuhan kekuasaan yang tinggi akan lebih sukses sebagai pemimpin daripada orang yang tidak menunjukkan pola – pola tersebut. Pemimpin yang mampu mendapatkan penghargaan, penghormatan, dan tanggapan yang tinggi daripada pengikutnya, sering kali digambarkan sebagai karismatik.

Disamping memiliki karakter – karakter dan motivasi yang berbeda dengan orang lain, adanya pengaruh gaya kepemimpinan terhadap efektivitas kepemimpinan, memberikan tambahan yang berharga terkait dengan bagaimana para pemimpin berperilaku dan bagaimana tindakan – tindakan mereka mempengaruhi para pengikut mereka.

2.11.2 Beberapa Definisi Kepemimpinan

Berikut ini beberapa definisi kepemimpinan :

a. Seseorang yang memimpin dan menunjukkan jalan

Seorang pemimpin harus mampu dan mau untuk berada di depan, dan memimpin jalannya. Ini memerlukan inisiatif, keberanian, dan iman.

b. Seseorang yang menunjukkan jalan, arah, dan komando

Seorang pemimpin tidak akan pernah dapat menyendiri. Ia adalah bagian dari kumpulannya dan bertanggung jawab sebagai pemimpinnya. Ia harus dapat diidentifikasi dengan kumpulannya itu, bukan sebagai anggota dari kelompoknya, melainkan sebagai penganggung jawab arah dan kegiatannya.

c. Seseorang yang memengaruhi sikap dan tindakannya dari yang lainnya

Seorang pemimpin memiliki kemampuan untuk memberikan inspirasi bagi yang lainnya pada tingkatan bahwa ia dengan tetap dan positif memengaruhi sikap serta tindakan mereka untuk bergerak ke arah yang benar. Ia menginspirasi mereka untuk tetap bergerak walaupun jalan sangat berat dan banyak rintangan di jalan itu.

d. Seseorang yang orang lain ingin mengikutinya

Seorang pemimpin memiliki karakter yang membuat orang lain ingin mengikutinya. Seorang pemimpin bukan seorang pemimpin bila tidak ada yang mengikutinya. Ini adalah tanda yang utama dari kualitas kepemimpinannya `orang mengikutinya`. Seseorang mungkin mengenal seluruh teori prinsip kepemimpinan. Mereka mungkin mampu untuk memberi kuliah atas hal – hal tersebut atau mungkin menulis tentang hal ini, tetapi tes dari kepemimpinan adalah kepengikutan. Apakah orang mengikutinya, dan hal apa yang telah atau sedang mereka hasilkan ?

e. Seseorang dengan kemampuan untuk memotivasi orang lain untuk menghasilkan

Bagian yang penting dari kepemimpinan yang efektif adalah kemampuan untuk memotivasi yang lain, misalnya membuat mereka bergerak ke arah yang benar. Namun, motivasi hanya efektif bila tujuan utamanya benar – benar tercapai. Beberapa pemimpin memang menggerakkan orang, mereka terlibat dalam segala kegiatan dan program, tetapi mereka tidak mencapai tujuan utama

mereka, hanya bergerak dalam berbagai kegiatan. Kepemimpinan yang benar menghasilkan yang tuntas. Ia mencapai target dan tujuan utamanya.

2.11.3 Prinsip – prinsip Kepemimpinan

Kepemimpinan adalah proses menggerakkan group dalam arah yang sama tanpa paksaan. Sukses kepemimpinan adalah kepemimpinan yang menciptakan langkah paling menari dari group untuk jangka panjang. Kepemimpinan adalah salah satu aspek penting dalam manajemen. Ada beberapa macam gaya kepemimpinan terhadap situasi yang berbeda, bisa digunakan sebagai model tingkah laku *leader* untuk menghadapi berbagai macam sifat anak buah yang dipimpinnya.

Kebanyakan orang percaya bahwa kekuasaan adalah inti dari kepemimpinan, dengan asumsi bahwa mereka memiliki pengetahuan dan kecerdasan. Namun, itu tidak otomatis benar. Para ilmuwan, peneliti, yang jenius, dan ahli filsafat adalah orang yang memiliki kemampuan untuk berpikir sangat tinggi sehingga melebihi batas pengetahuan, tetapi mereka memiliki kemampuan kepemimpinan sangat rendah. IQ bukanlah hal yang disamakan dengan kepemimpinan.

2.11.4 Kepemimpinan yang Dinamis

Kepemimpinan yang dinamis, yaitu corak kepemimpinan yang hanya mengenal istilah maju. Dia tidak pernah mengenal istilah mundur dan menyerah. Karakter kepemimpinan yang dinamis :

1. Positif

Dia terus bergerak maju menuju suatu sasaran yang pasti dengan membuat perencanaan – perencanaan yang mantap. Dia tidak menunggu sampai ada persoalan baru. Dia bereaksi untuk membuat rencana kerja dan langkah - langkah yang harus ditempuh untuk mencapai target. Sementara pemimpin yang tidak memiliki rencana kerja yang mantap, dia baru bereaksi untuk membuat rencana bila ada persoalan baru. Hal ini terjadi karena dia tidak memiliki rencana kerja yang mantap. Dan hanya sekedar menjalankan rutinitas. Tipe kepemimpinan ini tidak dinamis.

Karena memiliki rencana yang mantap, maka pemimpin yang berkarakter positif ini tidak mudah goyah dalam menghadapi kritikan – kritikan, tantangan – tantangan untuk mencapai target yang telah ditetapkan tersebut. Bahkan, tantangan – tantangan dan kritikan – kritikan tersebut tidak dianggap sebagai suatu beban atau penghalang, karena begitu sibuk dan memfokuskan perhatiannya untuk menuju ke sasaran sehingga semua tantangan atau kritikan – kritikan itu merupakan suatu kesulitan kecil yang akan dibawa lari pada sasaran.

2. Konstruktif

Dia bergerak menuju sasaran yang pasti dan tidak merugikan atau menghancurkan sekelilingnya, khususnya manusia. Jadi, dia tidak memiliki sifat destruktif, bahkan rekan – rekan yang bekerja bersama – sama dengan dia itu dibuatnya menjadi maju dan berprestasi juga.

Jangan menjadi pemimpin destruktif yang berdiri di atas puing – puing kehancuran rekan sekerja. Sebagai seorang pemimpin yang memiliki karakter konstruktif ini, apabila sudah mencapai sasaran, dia tidak diutamakan atau dipuji, justru rekan – rekan yang bekerja bersama – sama dengan dia yang lebih berprestasi dari dia. Kepemimpinan yang berkarakter konstruktif ini tidak bertepuk dada dan memuji diri sendiri kalau berhasil, dan tidak putus asa kalau dia gagal.

3. Kreatif

Dia bergerak maju menuju kepada rencana – rencana yang mantap, tidak menghancurkan atau merugikan rekan sekerjanya, dan kaya dengan kreasi – kreasi baru untuk mencapai sasaran. Kalau dia sudah membuat rencana yang mantap ini, namun ternyata rencana tersebut kemudian macet, maka dia tidak langsung putus asa. Dengan kreasinya yang baru, dia berusaha untuk mencari alternatif yang lain dengan konsep – konsep baru untuk meneruskan rencananya sehingga mencapai sasaran.

Meskipun benar adanya bahwa beberapa orang dilahirkan dengan bakat kepemimpinan yang hebat, yang alami dari yang lainnya, kemampuan untuk memimpin sebenarnya adalah terdiri dari kumpulan ketrampilan atau keahlian yang

hampir semuanya dapat dipelajari dan ditingkatkan. Namun, prosesnya tidak terjadi dalam semalam. Ketrampilan Kepemimpinan (*Leadership skill*) :

1. Memengaruhi
2. Memotivasi
3. Manajemen waktu
4. Komunikasi yang efektif
5. Menghargai dan men-*support*
6. Menciptakan ide dan semangat membangun

2.11.5 Gaya Kepemimpinan

Gaya kepemimpinan memiliki dua dimensi :

1. *Directive Behaviour* (Perilaku Mengarahkan) :

Perilaku pemimpin yang memberitahu dan menunjukkan kepada anggota tim apa yang harus dilakukan, kapan waktunya, bagaimana caranya, dan memberikan tanggapan atau hasil – hasil yang dicapai.

2. *Supportive Behaviour* (Perilaku Mendukung) :

Perilaku pemimpin yang memberikan pujian, mendengarkan, memberikan semangat, melibatkan anggota tim dalam mengambil keputusan.

Ada empat gaya yang merupakan kombinasi dari dua dimensi diatas :

1. *Directing* (Mengarahkan) – *High Directive Behaviour/Low Supportive*

- Pemimpin memberika instruksi khusus dalam hal peran dan sasaran, dan secara ketat mengikuti kinerja dari anggota tim dalam rangka memberikan masukan terhadap hasilnya.
- Mengarahkan berarti memberitahu orang apa yang harus dilakukan. Pemimpin harus menjelaskan semua tugas – tugas, memberikan informasi secara instruksi
- Gaya ini sangat sesuai untuk diterapkan pada para anggota yang baru atau yang belum berpengalaman. Gaya ini juga sangat membantu pada para anggota yang menemukan kesulitan untuk memulai sesuatu, atau pada saat mereka perlu bimbingan untuk menyusun aktivitasnya.

- Bahayanya atas gaya ini adalah tidak sadarnya si pemimpin berubah gaya dari pengarahan menjadi dominasi (*domination*).

2. *Coaching (Melatih) – High Directive Behaviour/High Supportive Behaviour*

- Pemimpin memberikan penjelasan – penjelasan, memancing usulan atau saran, memuji perilaku yang positif dan tetap mengarahkan penyelesaian tugas.
- Dalam gaya ini pemimpin menampung (mengakomodasi) ide – ide dari orang – orang, dan setelah itu baru dibuat keputusan.
- Pemimpin juga mungkin perlu membagi permasalahan yang dihadapi pada anggota atau mendengarkan, tetapi yang menentukan rencana tindakan tetap pada pemimpin.
- Gaya ini adalah suatu gaya yang baik untuk memecahkan permasalahan, tetapi sisi lainnya yang perlu diwaspadai adalah kecenderungan untuk terlalu terlibat dalam berbagai hal.

3. *Supporting (Mendukung) – High Supportive Behaviour/ Low Directive Behaviour*

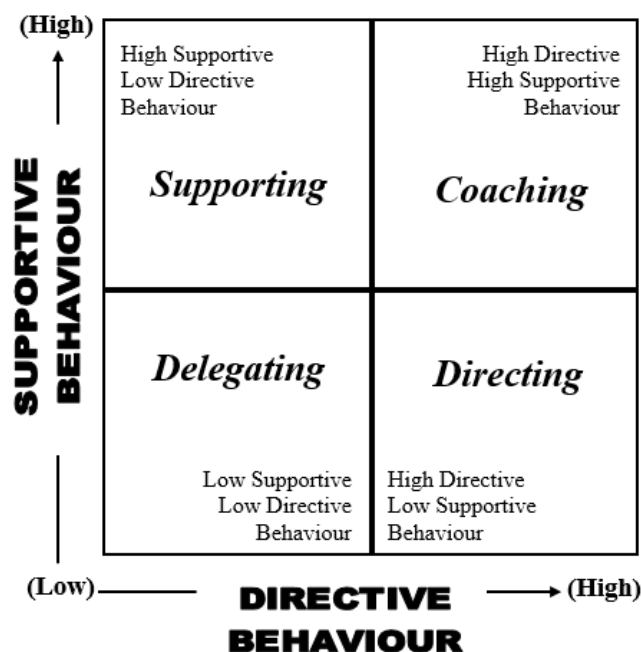
- Pemimpin dan anggota tim membuat keputusan bersama. Peran pemimpin adalah untuk menampung fasilitas, mendengarkan, memberikan umpan, semangat dan dukungan.
- Dalam memberikan dukungan, pemimpin memberikan dorongan untuk dilaksanakannya komunikasi ke atas, mencari ide – ide, mendengarkan, memberikan nasihat, memberikan pengakuan, dan membantu orang – orang yang dipimpin dalam membuat dan mengambil keputusan.
- Gaya ini akan membantu mengembangkan orang jika digunakan secara efektif.
- Pemimpin membimbing anggota untuk membuat keputusan yang baik. Selain itu, pemimpin menekankan bahwa anggota tim harus mengambil keputusan dengan bimbingannya.

- Sisi negatif gaya ini adalah kecenderungan untuk terlalu akomodatif.

4. *Delegating (Mendelegasikan) – Low Supportive Behaviour/Low Directive Behaviour*

- Pemimpin memberdayakan (*empower*) anggota tim untuk bertindak secara independen dan dengan dukungan yang sesuai untuk memastikan bahwa tugas terlaksana.
- Pemimpin memercayai secara penuh anggota yang dipimpinnya, anggota lebih mengetahui tentang suatu permasalahan dibandingkan dengan pemimpin, dan membiarkan anggota membuat keputusan – keputusan sendiri.
- Kebaikan gaya ini adalah rasa percaya untuk pendelegasian. Sisi negatifnya adalah sikap membiarkan anggota atau tidak mengacuhkannya.

Pemimpin yang luar biasa menggunakan keempat gaya kepemimpinan tersebut dalam rangka mencapai tujuan – tujuannya serta mendapatkan hasil yang maksimal dari orang – orang yang dipimpinnya.



Gambar 2.6 Diagram Empat Gaya Kepemimpinan

Gambar 2.6 Diagram Empat Gaya Kepemimpinan

2.13 Egois

Dalam serapan asing dan bahasa Indonesia, kata egois berarti orang yang mementingkan diri sendiri, tidak peduli akan orang lain atau masyarakat. Dalam kamus bahasa Indonesia online, egois berarti tingkah laku yang didasarkan atas dorongan untuk keuntungan diri sendiri dari pada untuk kesejahteraan orang lain atau segala perbuatan dan tindakan selalu disebabkan oleh keinginan untuk menguntungkan diri sendiri. Menurut bahasa egois berasal dari kata “ego” yang sebenarnya bukanlah sesuatu yang negative. Ego dapat memiliki makna sebagai “aku”; sebuah pribadi, diri sendiri, sebuah konsep individu tentang dirinya sendiri. Tidak ada yang negative dari kata ini. ego justru merupakan suatu langkah kebaikan dimana seorang individu sadar akan dirinya sendiri. Namun, ketika kata “ego” diberi akhiran –is dan menjadi “egois”, artinya menjadi individu yang mementingkan diri sendiri. Penelitian menunjukkan bahwa seorang yang pandai dan berkuasa cenderung memiliki sifat egois, karena memang jalan pikiran dialah yang dirasa benar. Ini kaitanya dengan pola pikir.

Sebenarnya, sikap egois itu tidak selalu menjurus kea rah yang negative, namun disaat sifat egois atau sifat ke “aku”an ini sangat besar, timbullah masalah-masalah. Pribadi egois adalah pribadi yang melihat segala sesuatu dari kacamataanya. Ia tidak bisa memahami pikiran orang, perasaan orang dan selalu menuntut orang untuk mengikuti pendapatnya. Pribadi egois juga pribadi yang mementingkan dirinya sendiri, dia tidak bisa mempertimbangkan kebutuhan orang, senantiasa mengedepankan kebutuhanya diatas kebutuhan orang lain. Jadi pribadi yang egois adalah pribadi yang susah sekali untuk tulus, sebab ujung-ujungnya untuk kepentinganya sendiri.

2.14 *Multi Objective Optimization*

Optimasi multi-tujuan (juga dikenal sebagai pemrograman multi-tujuan, optimasi vektor, optimasi multikriteria, optimasi multi attribute atau optimasi Pareto) adalah daerah beberapa pengambilan keputusan kriteria, yang berkaitan dengan masalah optimasi matematika yang melibatkan lebih dari satu fungsi tujuan yang akan dioptimalkan secara bersamaan. Optimasi multi-tujuan telah diterapkan di berbagai bidang ilmu pengetahuan, termasuk teknik, ekonomi dan logistik (lihat

bagian pada aplikasi untuk contoh rinci) dimana keputusan optimal perlu diambil dengan adanya trade-off antara dua atau lebih yang saling bertentangan tujuan. Meminimalkan penyerangan sekaligus memaksimalkan perlindungan terhadap pasukan dalam penelitian ini adalah contoh dari masalah optimasi multi-tujuan yang melibatkan dua dan tiga tujuan masing-masing. Dalam masalah-masalah praktis, bisa ada lebih dari tiga tujuan.

Para peneliti mempelajari masalah optimasi multi-tujuan dari sudut pandang yang berbeda dan, dengan demikian, terdapat filosofi solusi yang berbeda dan tujuan ketika mengatur dan pemecahannya. Tujuannya mungkin untuk menemukan satu set mewakili solusi optimal Pareto, dan / atau mengukur trade-off dalam memenuhi tujuan yang berbeda, dan / atau menemukan solusi tunggal yang memenuhi preferensi subjektif dari pengambil keputusan manusia (DM).

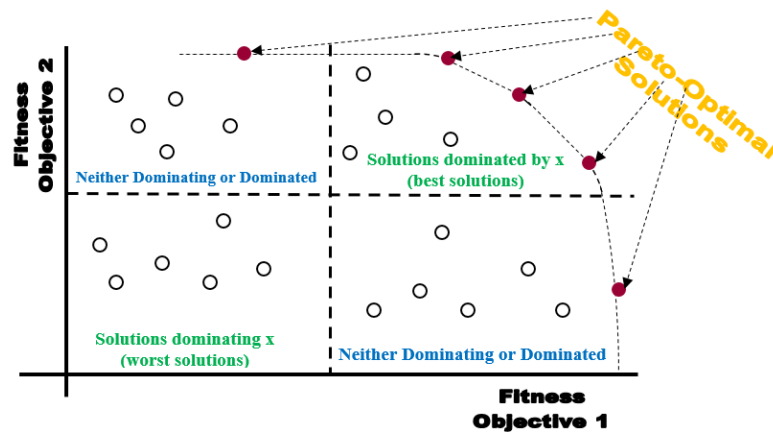
Multi Objective Optimization Problem (MOP) merupakan suatu proses yang secara bersamaan dalam mengoptimasi dua atau lebih *objective* (tujuan) yang berkonflik terhadap kendala – kendala tertentu [27]. MOP juga sering disebut dengan optimasi multi kriteria atau optimasi multi atribut. MOP dapat dijumpai pada bermacam – macam bidang, misalnya pada rancangan produk dan proses produksi, perdagangan jual – beli yang membutuhkan keputusan optimal dari *trade off* antara dua *objectif* atau lebih yang saling berkonflik. Contoh dari MOP meliputi pemaksimalan profit dan meminimalan biaya pada suatu produk, pemaksimalan penampilan dan meminimalan harga dari suatu produk bagi pembeli dan lain sebagainya. Pada MOP akan terdapat banyak variabel yang menjelaskan konstruksi dari system yang disebut juga dengan parameter – parameter dari permasalahan optimasi. Parameter tersebut adalah kombinasi dan tata letak dari komponen pada sebuah permasalahan rancangan sistem dan konfigurasi dari parameter – parameter tersebut ditujukan sebagai sebuah solusi untuk kerja dari suatu solusi terhadap permasalahan dapat dievaluasi untuk masing – masing *objective* yang dioptimalkan.

Optimasi merupakan tugas menemukan satu atau beberapa solusi yang sesuai untuk meminimalkan (atau memaksimalkan) satu atau lebih tujuan tertentu yang memenuhi semua dan kendala (jika ada). Optimasi persoalan single – *objective* melibatkan fungsi satu *objective* dan biasanya hasil dalam satu solusi,

dinamakan sebuah solusi optimal. Di sisi lain, tugas Optimasi Multi Objective mempertimbangkan beberapa konflik tujuan secara bersamaan. Dalam kasus ini biasanya tidak ada satu solusi optimal, tetapi satu set alternative dalam *trade off* berbeda, disebut Pareto Solusi Optimal atau solusi non-dominated. Meskipun adanya beberapa solusi pareto optimal, dalam praktiknya, biasanya hanya satu solusi yang akan dipilih. Dengan demikian pada optimasi Multiobjective, paling tidak ada dua tugas yang sama pentingnya, sebuah tugas optimasi untuk mencari solusi pareto yang optimal (melibatkan computer berbasis prosedur) dan sebuah keputusan tugas untuk memilih satu solusi yang paling disukai yang terakhir biasanya informasi keharusan pilihan dari pengambil keputusan (*Decision Maker – DM*). Tujuan penyelesaian dan pengaturan MOP adalah untuk menemukan suatu solusi untuk masing – masing dari objective yang telah dioptimasi dan mengkualifikasi bagaimana unggulnya solusi tersebut jika dibandingkan dengan solusi lainnya.

2.15 Pareto Optimal

Sebuah solusi bisa saja yang terbaik, terburuk dan juga tidak berhubungan terhadap solusi lainnya (baik dominating atau dominated) dengan tetap memperhatikan sebuah nilai objektif. Solusi terbaik dapat diartikan sebuah solusi tidak buruk pada objektif manapun dan paling tidak satu objektif lebih baik dari pada yang lainnya. Sebuah solusi optimal adalah solusi yang tidak didominasi oleh solusi manapun dalam ruang pencariannya. Seperti sebuah solusi optimal dikatakan Pareto – Optimal dan keseluruhan kumpulan dari seperti solusi optimal *trade-offs* dikatakan Pareto-optimal set. Sebagai bukti, dalam situasi yang nyata keputusan (*trade-off*) adalah proses yang diperlukan untuk mendapatkan solusi yang optimal. Walaupun terdapat beberapa cara untuk pendekatan persoalan multiobjektif optimization, sebagian besar pekerjaan terkonsentrasi pada perkiraan dari kumpulan pareto (Ajith Abrahan and Golberg 2005).



Gambar 2.7 Konsep Pareto Optimal

Multiobjective algoritma yang digunakan adalah NSGA II (Non Sort Genetic Algoritma II). Dasar dari NSGA II adalah proses eksekusi ranking sebelum operasi seleksi, Proses ini diidentifikasi sebagai solusi non dominan di dalam populasi, di tiap – tiap generasi ke bentuk *front nondominan*. Kemudian setelah itu adalah seleksi, *crossover*, dan mutasi. Didalam prosedur ranking yang pertama kali diidentifikasi adalah individu yang *non dominan* didalam populasi kemudian individu tersebut diasumsikan sebagai *front non dominan* yang pertama dengan nilai fitness yang paling besar untuk sementara [28]. Fitness yang nilainya sama diassign ke dalam *non dominan* tersebut untuk memperbaiki perbedaan. Didalam populasi digunakan metode *sharing* kemudian setelah itu individu dari *front* yang pertama sementara diabaikan kemudian populasi diproses lagi dengan cara yang sama untuk mengidentifikasi individual untuk *front nondominan* yang kedua. Sebuah nilai fitness sementara yang di*share* dari *front* sebelumnya yang diassign ke semua individu yang mempunyai *front* yang baru. Proses ini dilanjutkan sampai semua populasi terklarifikasi ke dalam *non dominan front*. Ketika *non dominan front* teridentifikasi kemudian populasi yang dihasilkan adalah menurut nilai fitness sementara tersebut.

NSGA pertama kali menggunakan *procedure stochastic remainder propotional selection* (SRS). Walaupun demikian kemungkinan juga bisa menggunakan berbagai teknik yang lain seperti *roulette wheel* atau *tournament* [N. Srinivas and Kalyanmoy Deb 1994]. Individu yang ada pada *front* yang pertama mempunyai nilai fitness yang paling tinggi. Metode ini difokuskan untuk mencari

daerah yang *non dominan* dan membantu untuk individu diatas daerahnya [Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan 2002]. Fitness *sharing* didalam algoritma genetika, teknik *sharing* tujuannya membuat formasi dan memperbaiki kestabilan dari sub populasi [Yau, Y. J.; Teo, J.; and Anthony 2007].

2.16 Algoritma Genetika

Metode adaptif yang sering digunakan untuk memecahkan masalah dalam pencarian nilai dalam sebuah optimasi adalah algoritma genetika. Algoritma genetika berdasarkan proses genetik pada makhluk hidup, dengan proses berkembangnya generasi dalam sebuah populasi yang secara lambat laun berdasarkan seleksi alam dalam kemampuan bertahan hidup (*survive*). Teori evolusi inilah yang digunakan dengan metode algoritma genetika untuk mencari solusi dari berbagai macam permasalahan yang diciptakan oleh J.H Holland [29].

2.15.1 Pengertian Algoritma Genetika [30]

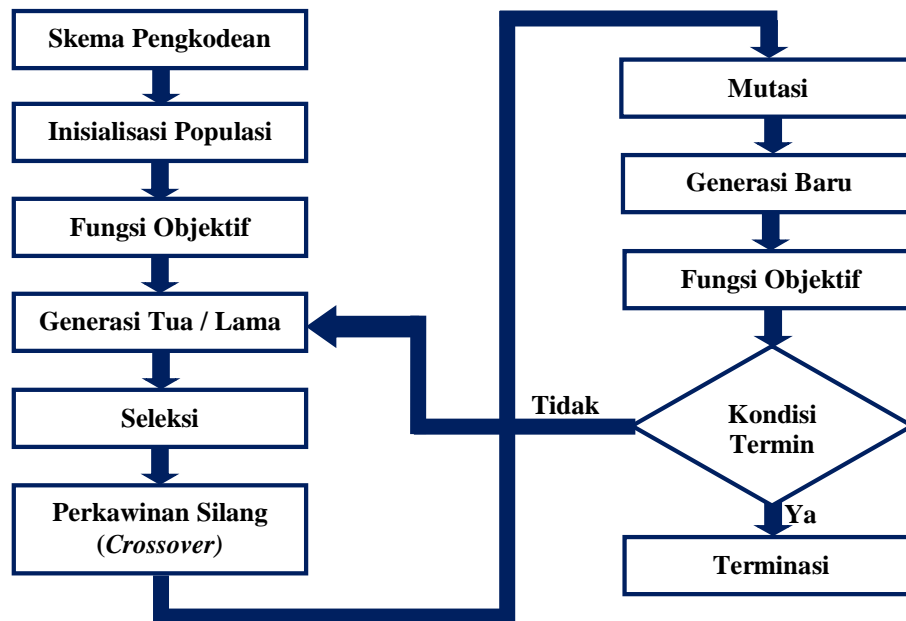
Algoritma ini ditemukan di Universitas Michigan, Amerika Serikat oleh John Holland (1975) melalui sebuah penelitian dan dipopulerkan oleh salah satu muridnya, David Goldberg (1989). Dimana mendefenisikan algoritma genetic ini sebagai metode algoritma pencarian berdasarkan pada mekanisme seleksi alam dan genetik alam. Algoritma genetik adalah algoritma yang berusaha menerapkan pemahaman mengenai evolusi alamiah pada tugas-tugas pemecahan-masalah (*problem solving*). Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik berikutnya yaitu pada suatu kondisi yang memaksimalkan kecocokannya atau lazim disebut *fitness*. Generasi ini akan merepresentasikan perbaikan-perbaikan pada populasi awalnya. Dengan melakukan proses ini secara berulang, algoritma ini diharapkan dapat mensimulasikan proses evolusioner. Pada akhirnya, akan didapatkan solusi-solusi yang paling tepat bagi permasalahan yang dihadapi. Untuk menggunakan algoritma genetik, solusi permasalahan direpresentasikan sebagai khromosom. Tiga aspek yang penting untuk penggunaan algoritma genetik:

1. Definisi fungsi fitness
2. Definisi dan implementasi representasi genetik
3. Definisi dan implementasi operasi genetik

Jika ketiga aspek di atas telah didefinisikan, algoritma genetik akan bekerja dengan baik. Tentu saja, algoritma genetik bukanlah solusi terbaik untuk memecahkan segala masalah. Sebagai contoh, metode tradisional telah diatur untuk mencari penyelesaian dari fungsi analitis convex yang “berperilaku baik” yang variabelnya sedikit. Pada kasus-kasus ini, metode berbasis kalkulus lebih unggul dari algoritma genetik karena metode ini dengan cepat menemukan solusi minimum ketika algoritma genetik masih menganalisa bobot dari populasi awal. Untuk problem-problem ini pengguna harus mengakui fakta dari pengalaman ini dan memakai metode tradisional yang lebih cepat tersebut. Akan tetapi, banyak persoalan realistik yang berada di luar golongan ini. Selain itu, untuk persoalan yang tidak terlalu rumit, banyak cara yang lebih cepat dari algoritma genetik. Jumlah besar dari populasi solusi, yang merupakan keunggulan dari algoritma genetik, juga harus mengakui kekurangannya dalam dalam kecepatan pada sekumpulan komputer yang dipasang secara seri-fitness function dari tiap solusi harus dievaluasi. Namun, bila tersedia komputerkomputer yang paralel, tiap prosesor dapat mengevaluasi fungsi yang terpisah pada saat yang bersamaan. Karena itulah, algoritma genetik sangat cocok untuk perhitungan yang paralel.

2.15.2 Struktur Umum Algoritma Genetika

Algoritma genetik memberikan suatu pilihan bagi penentuan nilai parameter dengan meniru cara reproduksi genetik, pembentukan kromosom baru serta seleksi alami seperti yang terjadi pada makhluk hidup. Algoritma Genetik secara umum dapat diilustrasikan dalam diagram alir berikut ini :



Gambar 2.8 Diagram Alur Algoritma Genetika

Golberg (1989) mengemukakan bahwa algoritma genetik mempunyai karakteristik-karakteristik yang perlu diketahui sehingga dapat terbedakan dari prosedur pencarian atau optimasi yang lain, yaitu:

1. Algoritma genetik dengan pengkodean dari himpunan solusi permasalahan berdasarkan parameter yang telah ditetapkan dan bukan parameter itu sendiri.
2. Algoritma genetik pencarian pada sebuah solusi dari sejumlah individu-individu yang merupakan solusi permasalahan bukan hanya dari sebuah individu.
3. Algoritma genetik informasi fungsi objektif (fitness), sebagai cara untuk mengevaluasi individu yang mempunyai solusi terbaik, bukan turunan dari suatu fungsi.
4. Algoritma genetik menggunakan aturan-aturan transisi peluang, bukan aturan-aturan deterministik.

Variabel dan parameter yang digunakan pada algoritma genetik adalah:

1. Fungsi fitness (fungsi tujuan) yang dimiliki oleh masing-masing individu untuk menentukan tingkat kesesuaian individu tersebut dengan kriteria yang ingin dicapai.

2. Populasi jumlah individu yang dilibatkan pada setiap generasi.
3. Probabilitas terjadinya persilangan (crossover) pada suatu generasi.
4. Probabilitas terjadinya mutasi pada setiap individu.
5. Jumlah generasi yang akan dibentuk yang menentukan lama penerapan algoritma genetik.

Secara umum struktur dari suatu algoritma genetik dapat mendefinisikan dengan langkah-langkah sebagai berikut:

1. **Membangkitkan populasi awal**

Populasi awal ini dibangkitkan secara random sehingga didapatkan solusi awal. Populasi itu sendiri terdiri atas sejumlah kromosom yang merepresentasikan solusi yang diinginkan.

2. **Membentuk generasi baru**

Untuk membentuk generasi baru, digunakan operator reproduksi/ seleksi, crossover dan mutasi. Proses ini dilakukan berulang-ulang sehingga didapatkan jumlah kromosom yang cukup untuk membentuk generasi baru dimana generasi baru ini merupakan representasi dari solusi baru. Generasi baru ini dikenal dengan istilah anak (offspring).

3. **Evaluasi solusi**

Pada tiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang dinamakan fitness. Nilai fitness suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai fitness setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Bila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah 2. Beberapa kriteria berhenti sering digunakan antara lain: berhenti pada generasi tertentu, berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai fitness tertinggi tidak berubah, berhenti dalam n generasi tidak didapatkan nilai fitness yang lebih tinggi.

2.15.3 Pengkodean

Pengkodean adalah suatu teknik untuk menyatakan populasi awal sebagai calon solusi suatu masalah ke dalam suatu kromosom sebagai suatu kunci pokok persoalan ketika menggunakan algoritma genetik. Berdasarkan jenis symbol yang digunakan sebagai nilai suatu gen, metode pengkodean dapat diklasifikasikan sebagai berikut:

2.15.3.1 Pengkodean biner

Merupakan cara pengkodean yang paling umum digunakan karena adalah yang pertama kali digunakan dalam algoritma genetik oleh Holland. Keuntungan pengkodean ini adalah sederhana untuk diciptakan dan mudah dimanipulasi. Pengkodean biner memberikan banyak kemungkinan untuk kromosom walaupun dengan jumlah nilai-nilai yang mungkin terjadi pada suatu gen yang sedikit (0 dan 1). Di pihak lain, pengkodean biner sering tidak sesuai untuk banyak masalah dan kadang pengoreksian harus dilakukan setelah operasi crossover dan mutasi.

Tabel 2.2 Contoh Pengkodean Biner

Kromosom A	1 0 1 1 0 0 1 0 1 1 0 0 0 1 0 1 1 1 0 0 0 1
Kromosom B	1 1 0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1

2.15.3.2 Pengkodean Bilangan Riil atau Pengkodean Permutasi

Suatu pengkodean bilangan dalam bentuk riil. Masalah optimalisasi fungsi dan optimalisasi kendala lebih tepat jika diselesaikan dengan pengkodean bilangan riil karena struktur topologi ruang genotif untuk pengkodean bilangan riil identik dengan ruang fenotifnya, sehingga mudah membentuk operator genetik yang efektif dengan cara memakai teknik yang dapat digunakan yang berasal dari metode konvensional.

Tabel 2.3 Contoh Pengkodean Bilangan Riil atau Bilangan Permutasi

Kromosom A	1 5 3 2 6 4 7 9 8
Kromosom B	8 5 6 7 2 3 1 4 9

2.15.3.3 Pengkodean Nilai

Pengkodean ini menyatakan setiap kromosom berupa string nilai aslinya. Nilai tersebut dapat berupa angka, huruf, atau kombinasi keduanya, disesuaikan dengan kasusnya. Metode ini lebih efektif dan tampak alami karena sesuai kasus yang diselesaikan.

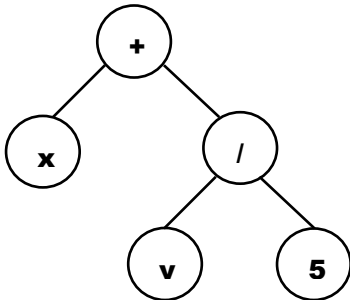
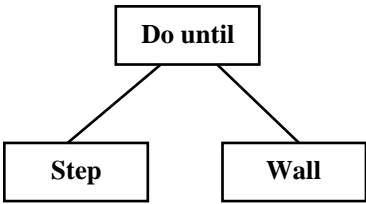
Tabel 2.4 Contoh Pengkodean Nilai

Kromosom A	1.2324 5.3243 0.4556 2.3293 2.4545
Kromosom B	ABDJEIFJDHDIERJFDLDFLFEGT
Kromosom C	(back), (back), (right), (forward), (left)

2.15.3.4 Pengkodean Struktur Data atau Pengkodean Pohon

Model pengkodean yang menggunakan struktur data. Pengkodean ini digunakan untuk masalah kehidupan yang lebih kompleks seperti perencanaan jalur robot, dan masalah pewarnaan Graph. Pada pengkodean ini kromosom diumpamakan sebagai setiap cabang dari berbagai obyek seperti fungsi dan perintah dalam bahasa pemrograman.

Tabel 2.5 Contoh Pengkodean Cabang

Kromosom A	Kromosom B
	
(+ x (/ 5y))	(do_until step wall)

2.15.4 Fungsi Evaluasi Kebugaran (*fitness*)

Fungsi kebugaran menggambarkan hasil atau solusi yang telah dikodekan. Selama proses GA induk harus digunakan *crossover* dan mutase untuk memperoleh

keturunan. Jika GA dirancang dengan baik populasi akan mengalami sebuah konvergensi dan mengarah pada solusi yang optimum.

Pada permulaan optimasi, biasanya nilai kebugaran akan memiliki rentang yang lebar. Seiring dengan bertambahnya generasi beberapa kromosom mendominasi populasi dan mengakibatkan rentang nilai kebugaran makin kecil. Hal ini dapat mengakibatkan konvergensi dini. Dalam hal ini hanya proses mutasi yang mampu menghasilkan kromosom yang relative baru dan merupakan cara untuk menghindari kromosom tertentu mendominasi populasi.

2.15.5 Operator Genetik

Algoritma genetik merupakan proses pencarian yang heuristik dan acak sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan algoritma genetik dalam menemukan solusi optimum suatu masalah yang diberikan. Hal yang harus diperhatikan adalah menghindari terjadinya konvergensi premature, yaitu mencapai solusi optimum yang belum waktunya, dalam arti bahwa solusi yang diperoleh adalah hasil optimum lokal. Operator genetik yang digunakan setelah proses evaluasi tahap pertama membentuk populasi baru dari generasi sekarang. Operator-operator tersebut adalah operator seleksi, crossover dan mutasi.

2.15.5.1 Seleksi

Seleksi bertujuan memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling fit. Langkah pertama dalam seleksi ini adalah pencarian nilai fitness. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai fitness inilah yang nantinya akan digunakan pada tahap seleksi berikutnya (Kusumadewi, 2003).

Kemampuan algoritma genetik untuk memproduksi kromosom yang lebih baik secara progresif tergantung pada penekanan selektif (*selective pressure*) yang diterapkan ke populasi. Penekanan selektif dapat diterapkan dalam dua cara. Cara pertama adalah membuat lebih banyak kromosom anak yang dipelihara dalam

populasi dan memilih hanya kromosom-kromosom terbaik bagi generasi berikut. Walaupun orang tua dipilih secara acak, metode ini akan terus menghasilkan kromosom yang lebih baik berhubungan dengan penekanan selektif yang diterapkan pada individu anak tersebut.

Cara lain menerapkan penekanan selektif adalah memilih orang tua yang lebih baik ketika membuat keturunan baru. Dengan metode ini, hanya kromosom sebanyak yang dipelihara dalam populasi yang perlu dibuat bagi generasi berikutnya. Walaupun penekanan selektif tidak diterapkan ke level keturunan, metode ini akan terus menghasilkan kromosom yang lebih baik, karena adanya penekanan selektif yang diterapkan ke orangtua. Ada beberapa metode untuk memilih kromosom yang sering digunakan antara lain adalah seleksi roda rolet (roulette wheel selection), seleksi ranking (rank selection) dan seleksi turnamen (tournament selection). Metode seleksi yang sering digunakan yaitu turnamen dan mesin *roulette* [31]. Dan dalam penelitian ini, seleksi yang digunakan adalah seleksi turnamen. Berikut masing – masing penjelasannya :

a. Turnamen

Metode turnamen ini diawali dengan menetapkan suatu nilai random untuk individu - individu yang disebut nilai *tour*. Parameter yang digunakan pada metode ini adalah ukuran *tour* yang bernilai antara 2 sampai N (dimana N adalah jumlah individu dalam suatu populasi).

b. Mesin roulette

Metode mesin *roulette* ini paling banyak digunakan karena merupakan metode yang paling sederhana atau sering disebut dengan *stochastic sampling with replacement*. Alur metode mesin *roulette* ini adalah sebagai berikut:

1. Masing-masing individu dihitung nilai *fitness*nya (f_i adalah nilai *fitness* ke- i , untuk i adalah individu ke-1 s/d ke- n).
2. Dihitung total *fitness* semua individu dalam satu populasi.
3. Dihitung probabilitas masing-masing individu dalam satu populasi.
4. Dari probabilitas tersebut, dibangkitkan nilai random untuk masing – masing individu misalnya nilai random angka 1 sampai 100.

Ditentukan individu yang terpilih dalam proses seleksi berdasarkan hasil pembangkitan bilangan random.

2.15.5.2 Crossover

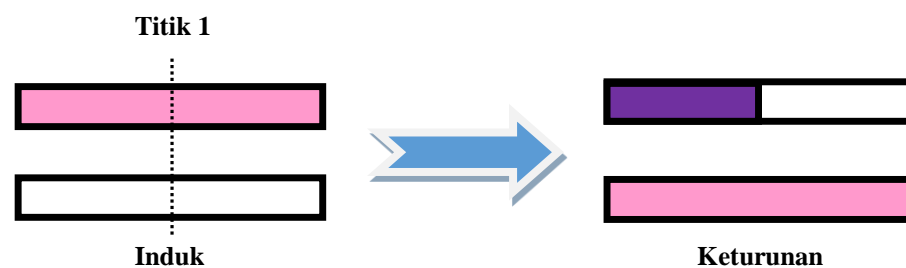
Crossover (perkawinan silang) bertujuan menambah keanekaragaman string dalam populasi dengan penyilangan antar-string yang diperoleh dari sebelumnya. Beberapa jenis crossover tersebut adalah:

1. Crossover 1-titik

Pada crossover dilakukan dengan memisahkan suatu string menjadi dua bagian dan selanjutnya salah satu bagian dipertukarkan dengan salah satu bagian dari string yang lain yang telah dipisahkan dengan cara yang sama. Proses yang demikian dinamakan operator crossover satu titik seperti diperlihatkan pada gambar dan tabel berikut :

Tabel 2.6 Tabel *Crossover* 1-Titik

Kromosom Orang Tua 1	1 1 0 0 1 0 1 1
Kromosom Orang Tua 2	1 1 0 1 1 1 1 1
Keturunan	1 1 0 0 1 1 1 1



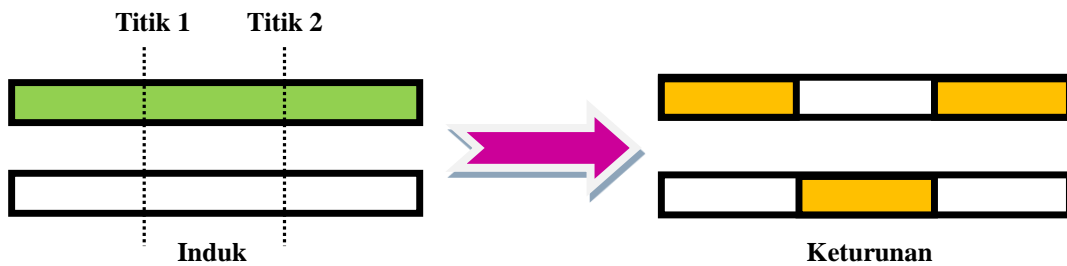
Gambar 2.9 Ilustrasi Operator Dengan Satu Titik Persilangan

2. Crossover 2-titik

Proses crossover ini dilakukan dengan memilih dua titik crossover. Kromosom keturunan kemudian dibentuk dengan barisan bit dari awal kromosom sampai titik crossover pertama disalin dari orangtua pertama, bagian dari titik crossover pertama dan kedua disalin dari orangtua kedua, kemudian selebihnya disalin dari orangtua pertama lagi.

Tabel 2.7 Tabel *Crossover* 2-Titik

Kromosom Orang Tua 1	1 1 0 0 1 0 1 1
Kromosom Orang Tua 2	1 1 0 1 1 1 1 1
Keturunan	1 1 0 1 1 1 1 1



Gambar 2.10 Ilustrasi Operator Dengan Dua Titik Persilangan

3. Crossover Seragam

Crossover seragam menghasilkan kromosom keturunan dengan menyalin bit-bit secara acak dari kedua orangtuanya.

Tabel 2.8 Tabel *Crossover* Seragam

Kromosom Orang Tua 1	1 1 0 0 1 0 1 1
Kromosom Orang Tua 2	1 1 0 1 1 1 1 1
Keturunan	1 1 0 1 1 1 1 1

2.15.5.3 Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Operasi crossover yang dilakukan pada kromosom dengan tujuan untuk memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi mendatang dengan fitness yang lebih baik, dan lama-kelamaan menuju solusi optimum yang diinginkan. Akan tetapi, untuk mencapai hal ini, penekanan selektif juga memegang peranan yang penting. Jika dalam proses pemilihan kromosom-kromosom cenderung pada kromosom yang memiliki fitness yang tinggi saja, konvergensi premature, yaitu mencapai solusi yang optimal lokal sangat mudah terjadi. Untuk menghindari konvergensi premature tersebut dan tetap menjaga perbedaan (diversity) kromosom-kromosom dalam populasi, selain

melakukan penekanan selektif yang lebih efisien, operator mutasi juga dapat digunakan.

Proses mutasi dalam system biologi berlangsung dengan mengubah isi allele gen pada suatu locus dengan allele yang lain. Proses mutasi ini bersifat acak sehingga tidak selalu menjamin bahwa setelah proses mutasi akan diperoleh kromosom dengan fitness yang lebih baik. Operator mutasi merupakan operasi yang menyangkut satu kromosom tertentu. Beberapa cara operasi mutasi diterapkan dalam algoritma genetic menurut jenis pengkodean terhadap phenotype, antara lain:

1. Mutasi dalam Pengkodean Biner

Mutasi pada pengkodean biner merupakan operasi yang sangat sederhana. Proses yang dilakukan adalah menginversi nilai bit pada posisi tertentu yang terpilih secara acak (atau menggunakan skema tertentu) pada kromosom, yang disebut *inverse bit*.

Tabel 2.9 Contoh Mutasi pada Pengkodean Biner

Kromosom Sebelum Mutasi	1 0 0 1 0 1 1 1
Kromosom Setelah Mutasi	1 0 0 1 0 0 1 1

2. Mutasi dalam Pengkodean Permutasi

Proses mutasi yang dilakukan dalam pengkodean biner dengan mengubah langsung bit-bit pada kromosom tidak dapat dilakukan pada pengkodean permutasi karena konsistensi urutan permutasi harus diperhatikan. Salah satu cara yang dapat dilakukan adalah dengan memilih dua posisi (locus) dari kromosom dan kemudian nilainya saling dipertukarkan.

Tabel 2.10 Contoh Mutasi pada Pengkodean Permutasi

Kromosom Sebelum Mutasi	1 2 3 4 6 5 8 7 9
Kromosom Setelah Mutasi	1 2 7 4 6 5 8 3 9

3. Mutasi dalam Pengkodean Nilai

Mutasi pada pengkodean nilai hampir sama dengan yang dilakukan pada pengkodean biner, tetapi yang dilakukan bukan menginversi nilai bit. Penerapannya bergantung pada jenis nilai yang digunakan. Sebagai contoh untuk nilai riil, proses

mutasi dapat dilakukan seperti yang dilakukan pada pengkodean permutasi, dengan saling mempertukarkan nilai dua gen pada kromosom.

4. Mutasi dalam Pengkodean Pohon

Mutasi dalam pengkodean pohon dapat dilakukan antara lain dengan cara mengubah operator (+, -, *, /) atau nilai yang terkandung pada suatu verteks pohon yang dipilih. Atau, dapat juga dilakukan dengan memilih dua verteks dari pohon dan saling mempertukarkan operator atau nilainya.

Tidak setiap gen selalu dimutasi tetapi mutasi dikontrol dengan probabilitas tertentu yang disebut dengan mutation rate (probabilitas mutasi) dengan notasi P_m . Jenis operator mutasi antara lain :

a. Mutasi Terarah

Mutasi terarah tergantung dari informasi gen. Informasi gen tersebut berupa nilai pelanggaran gen (violation gen). Ini berarti bahwa setiap gen mempunyai peluang yang berbeda untuk terjadi mutasi. Gen yang mempunyai nilai pelanggaran yang lebih besar maka gen tersebut mempunyai peluang untuk terjadi mutasi. Mutasi ini menghubungkan nilai pelanggaran relatif (nilai pelanggaran suatu gen dibagi dengan nilai pelanggaran total suatu kromosom) dengan probabilitas terjadinya mutasi dari suatu gen pada kromosom. Hubungan tersebut dinyatakan secara matematis sebagai berikut :

$$nr(i) = \frac{n(i)}{1 + n_{total}}$$

$$pm(i) = (1 + nr(i))^2 pm$$

Keterangan persamaan :

$nr(i)$: nilai pelanggaran relative gen ke-i

n_{total} : nilai pelanggaran total kromosom

$pm(i)$: probabilitas mutasi gen ke-i

pm : probabilitas mutasi

b. Mutasi Biasa

Mutasi biasa tidak tergantung dari informasi gen. Setiap gen mempunyai peluang yang sama untuk terjadi mutasi.

2.15.5.4 Elitisme

Selain kedua operator diatas (pindah silang dan mutasi), masih ada satu lagi operasi yang dapat dilakukan dengan Algoritma Genetika, yaitu *elitism*. Operator ini bertujuan untuk menghilangkan kromosom terburuk dari satu generasi dan menggantikannya dengan kromosom terbaik dari generasi sebelumnya. Elitisme ini adalah proses pengopian satu atau lebih individu yang bernilai fitness tinggi agar tidak hilang selama evolusi. Ini dilakukan karena seleksi dilakukan secara acak, maka tidak ada jaminan bahwa suatu individu bernilai fitness tertinggi akan selalu terpilih. Jika terpilih mungkin individu tersebut akan rusak karena proses pindah silang (crossover). Prosedur ini digunakan pada Algoritma Genetik berjenis generational replacement.

2.15.5.5 Parameter Algoritma Genetika

Pengoperasian algoritma genetik dibutuhkan 4 parameter (Juniawati, 2003) yaitu :

1. Probabilitas Persilangan (Crossover Probability) $\rightarrow (P_c)$

Menunjukkan kemungkinan crossover terjadi antara 2 kromosom. Jika tidak terjadi crossover maka keturunannya akan sama persis dengan kromosom orangtua, tetapi tidak berarti generasi yang baru akan sama persis dengan generasi yang lama. Jika probabilitas crossover 100% maka semua keturunannya dihasilkan dari crossover. Crossover dilakukan dengan harapan bahwa kromosom yang baru akan lebih baik. Probabilitas ini digunakan untuk mengendalikan operator pindah silang, Dalam hal ini, dalam populasi terdapat $P_c \times \text{POPSIZE}$ struktur (individu) yang melakukan pindah silang. Semakin besar nilai probabilitas pindah silang maka semakin cepat struktur baru diperkenalkan dalam populasi. Namun jika probabilitas pindah silang terlalu besar maka struktur dengan nilai obyektif yang baik dapat hilang dengan lebih cepat dari seleksi. Sebaliknya probabilitas pindah silang kecil

maka akan menghalangi proses pencarian dalam GA. Menurut [Zbigniew Michalewics 1996] banyak aplikasi GA yang menggunakan nilai probabilitas pindah silang antara 0.65 – 1.

2. Probabilitas Mutasi (Mutation Probability) → (P_m)

Menunjukkan kemungkinan mutasi terjadi pada gen-gen yang menyusun sebuah kromosom. Jika tidak terjadi mutasi maka keturunan yang dihasilkan setelah crossover tidak berubah. Jika terjadi mutasi bagian kromosom akan berubah. Jika probabilitas 100%, semua kromosom dimutasi. Jika probabilitasnya 0%, tidak ada yang mengalami mutasi. Mutasi digunakan untuk meningkatkan variasi populasi. Probabilitas mutasi ini digunakan untuk menentukan tingkat mutasi yang terjadi, karena frekwensi terjadinya mutasi tersebut menjadi $P_m \times \text{POPSIZE} \times N$, dimana N adalah panjang struktur atau gen dalam satu individu. Probabilitas yang rendah akan menyebabkan gen – gen yang berpotensi tidak dicoba. Dan sebaliknya tingkat mutasi yang tinggi akan menyebabkan keturunan akan semakin mirip dengan induknya. Dalam GA mutasi menjalankan aturan penting yaitu :

- a. Mengganti gen yang hilang selama proses seleksi
- b. Menyediakan gen – gen yang tidak muncul pada saat inisialisasi alam populasi.

Banyak aplikasi GA yang menggunakan angka probabilitas antara 0.00 – 0.1

3. Jumlah Individu

Menunjukkan jumlah kromosom yang terdapat dalam populasi (dalam satu generasi). Jika hanya sedikit kromosom dalam populasi maka algoritma genetik akan mempunyai sedikit variasi kemungkinan untuk melakukan crossover antara orangtua karena hanya sebagian kecil dari search space yang dipakai. Sebaliknya jika terlalu banyak maka algoritma genetik akan berjalan lambat.

4. Jumlah Populasi

Menentukan jumlah populasi atau banyaknya generasi yang dihasilkan, digunakan sebagai batas akhir proses seleksi, persilangan dan mutasi.

5. Panjang Kromosom

Panjang kromosom berbeda – beda sesuai dengan model permasalahan. Titik solusi dari ruang permasalahan dikodekan dalam bentuk kromosom / string yang terdiri dari komponen genetic terkecil yaitu gen. Pengkodean dapat memakai bilangan seperti string biner, integer, floating point dan abjad.

Mekanisme secara umum dari GA digambarkan sebagai proses kerjanya diawali dengan inisialisasi satu rangkaian nilai random yang disebut dengan populasi. Setiap individu dalam satu populasi dinamakan kromosom. Setiap individu dievaluasi dengan menggunakan nilai kebugaran atau *fitness*. Untuk menghasilkan generasi selanjutnya sebagai individu baru yang disebut dengan *offspring*, dibentuk melalui persilangan dua individu dengan menggunakan operator pindah silang (*crossover*) dan memodifikasi sebuah kromosom atau disebut juga dengan mutasi.

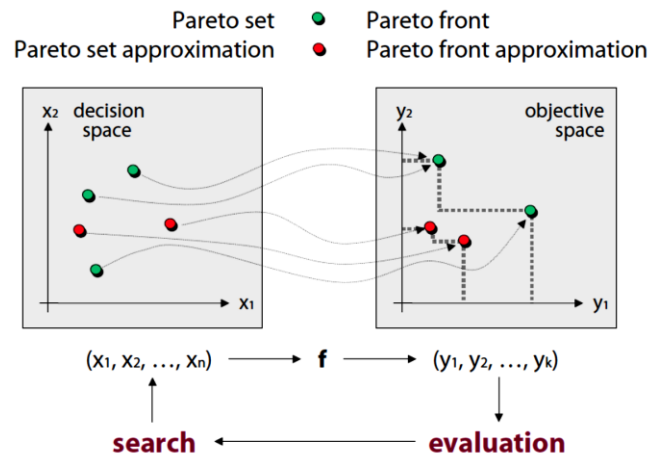
2.17 Algoritma NSGA-II

Non-dominated sorting GA(NSGA) diperkenalkan oleh Srinivas dan Deb pada tahun 1994 [N. Srinivas and Kalyanmoy Deb 1994]. NSGA [32] adalah algoritma non-dominasi populer berbasis genetik untuk optimasi multi obyektif. Ini adalah algoritma yang sangat efektif tetapi telah umum dikritik karena kompleksitas komputasi, kurangnya elitisme dan untuk memilih nilai parameter optimal untuk parameter berbagi σ_{share} . Sebuah versi modifikasi, NSGA- II [28] dikembangkan, yang memiliki algoritma sorting yang lebih baik, menggabungkan elitisme dan tidak ada parameter berbagi perlu dipilih secara prioritas. Walaupun telah disebutkan diawal ada beberapa nilai *critism* dari NSGA.

Populasi diinisialisasi seperti biasa. Setelah penduduk di diinisialisasi populasi diurutkan berdasarkan non-dominasi dalam setiap depan. Makhluk depan pertama yang sama sekali non-dominan dalam populasi saat ini dan makhluk depan kedua didominasi oleh individu dalam front pertama saja dan depan berjalan seterusnya. Masing Masing individu dalam setiap depan ditetapkan peringkat (fitness) nilai atau berdasarkan depan yang mereka miliki. Individu di depan pertama diberi nilai fitness dari 1 dan individu dalam kedua ditetapkan nilai fitness sebagai 2 dan seterusnya. Selain nilai fitness parameter baru yang disebut *crowding*

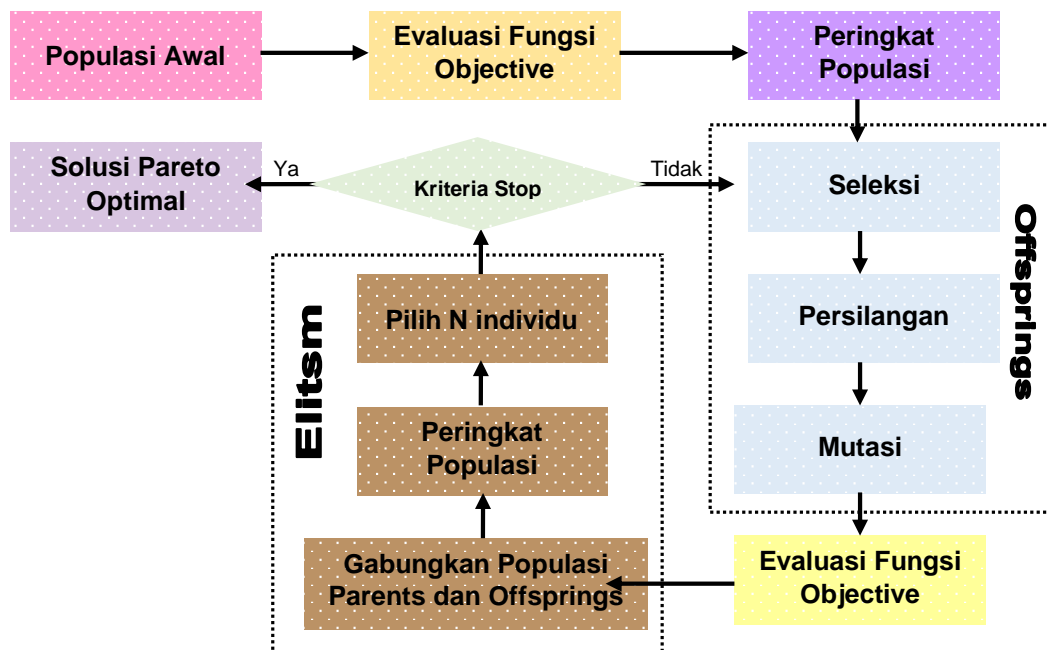
distance adalah menghitung setiap individu. *Crowding distance* adalah ukuran dari seberapa dekat individu dengan tetangganya. Besar rata-rata *crowding distance* akan menghasilkan keragaman yang lebih baik dalam populasi. Orang tua yang dipilih dari populasi dengan menggunakan turnamen seleksi biner berdasarkan pangkat dan *crowding distance*. Seorang individu yang dipilih dalam peringkat ini lebih rendah dari yang lain atau jika *crowding distance* lebih besar dari yang lain. Yang dipilih adalah populasi yang menghasilkan keturunan dari crossover dan operator mutasi, yang akan dibahas secara rinci pada bagian selanjutnya. Populasi dengan populasi saat ini dan keturunan saat ini diurutkan lagi berdasarkan non-dominasi dan hanya yang terbaik N individu yang dipilih, di mana N adalah ukuran populasi. Pemilihan ini didasarkan pada pangkat dan pada *crowding distance* di bagian depan sebelumnya.

Algoritma Genetika yang telah dijelaskan pada sub bab sebelumnya merupakan metode optimasi dengan obyektif tunggal. Metode tersebut digunakan untuk masalah optimasi skalar karena fungsi obyektif selalu mencapai nilai optimal global tunggal atau skalar. Untuk multi obyektif, fungsi multi-obyektif membentuk vektor sehingga biasa disebut dengan optimasi vektor [33]. Permasalahan optimasi multi obyektif tidak seperti permasalahan optimasi obyektif tunggal, tidak membutuhkan solusi optimal yang meminimalisasi semua fungsi multi obyektif secara simultan. Seringkali terjadi konflik antara obyektif satu dengan yang lain dan parameter optimal dari obyektif biasanya tidak mengarahkan kepada optimalitas obyektif lainnya (terkadang membuat lebih buruk). Sebagai contoh, sebuah keluarga yang menginginkan pelayanan kelas satu saat liburannya, namun menginginkan harga yang murah. Pelayanan tingkat tinggi (satu obyektif) akan sangat memakan biaya, namun ini menyebabkan konflik dengan obyektif lainnya (untuk meminimalisir biaya) [34].



Gambar 2.11 Ruang Keputusan dan Obyektif (*Decision and Objective Space*) [34]

Langkah utama pada NSGA-II adalah bahwa n *offspring* (S_t) diciptakan dari n *parent* (P_t) menggunakan algoritma genetika standar. Populasi keseluruhan (W_t) berukuran $2n$ dibentuk dengan menggabungkan S_t dan P_t ($W_t = P_t \cup S_t$). Kemudian sorting nondominasi diterapkan kepada W_t . Kemudian populasi baru akan terisi, satu kali dalam satu waktu, oleh solusi nondominasi yang berbeda. Karena populasi dari W_t adalah $2n$, hanya separuh dari jumlah tersebut yang akan menjadi populasi baru, memilih solusi nondominasi dari Pareto front dengan keragaman tinggi dan membuang solusi yang tersisa.



Gambar 2.12 Blok Diagram NSGA-II [34]

2.16.1 Inisialisasi Populasi

Inisialisasi populasi ini merupakan langkah awal algoritma NSGA-II. Populasi diinisialisasi berdasarkan range dari permasalahan atau parameter – parameter yang lain jika itu ada.

2.16.2 Non-Dominated Sort

Inisialisasi populasi yang di-sort berdasarkan non itness algoritma digambarkan dibawah ini :

Algoritma 2.1 Algoritma Non-Dominated Sort

1. Individu p pada populasi P akan diolah sebagai berikut :
 - a. Buat kumpulan $S_p = \{ \}$ berisi nama kumpulan individu yang itness dengan p
 - b. Buat counter $a_p = 0$ sebagai penghitung jumlah dari individu yang itness di $S_p = \{ \}$
 - c. Misal untuk individu q di P
 - Jika p didominasi q maka
Tambahkan q ke himpunan S_p i.e $S_p = S_p \cup \{q\}$
 - Selain itu jika q itness di p then
Increment itness hitung p i.e $n_p = n_p + 1$
 - d. if $n_p = 0$, tidak ada individu yang itness di p then p adalah *front* pertama
 $p_{rank} = 1$ update *front* pertama dengan menambah p ke *front* 1 $F_1 = F_1 \cup \{p\}$
2. Kemudian akan dibawa untuk semua individu didalam populasi P
3. Inisialisasi penghitung *front* ke 1 $i = 1$
4. Ketika i_{th} *front* tidak kosong $F_i \neq \emptyset$
 - $Q = \{ \}$ untuk menyimpan individu dari $(i + 1)_{th}$ front.
 - For each individu q di S_p (S_p adalah kumpulan dari individu yang didominasi p)
 - * $N_q = n_q - 1$, *decrement* penghitungan dominasi untuk individu q.

- * if $n_q=0$ then tidak ada individu dalam subsequent *front* akan didominasi q . kemudian $q_{rank} = i + 1$. Update himpunan Q dengan individu q $Q = Q \cup q$
- Increment *front counter* dengan 1
- Sekarang himpunan Q adalah *front* dan $F_i = Q$

2.16.3 Crowding Distance

Crowding distance dapat dihitung sebagai berikut :

Algoritma 2.2 Algoritma Crowding Distance

1. For each front F_i , n adalah jumlah dari individu
 - a. Inisialisasi jarak menjadi 0 untuk semua individu $F_i(d_j)$ dimana j berelasi ke individu j^{th} di front F_i
 - b. For each fungsi objektif m
2. Sort individu di front F_i berdasarkan objektif m
 $I = \text{sort}(F_i, m)$
3. Menentukan jarak ke nilai batas untuk masing – masing individu F_i $I(d_1) = \sim$ dan $I(d_n) = \sim$
4. For $k = 2$ to $(n-1)$
5. $I(k).m$ adalah nilai dari M^{th} fungsi objektif dari individu k^{th} di I

Ide dasar crowding distance adalah menemukan jarak Euclidian diantara masing – masing individu di sebuah front berdasarkan nilai m objektifnya didalam dimensi ruang m jarak pada individu di pilih ketika jarak itu ditentukan.

2.16.4 Selection

Seleksi dilakukan menggunakan *Crowed-comparation-operator* (\prec_n). Perbandingan dilakukan berdasarkan non domination P_{rank} dan *crowding distance*. Algoritmanya sebagai berikut :

Algoritma 2.3 Algoritma Slection

1. Non-domination rank P_{rank} individu di front F_i akan mempunyai rank sebagai $P_{rank} = i$
2. Crowding distance $F_i(d_j)$

- a. $P <_n q$ if
 - i. $P_{rank} < q_{rank}$
 - ii. Atau if p dan q mempunyai front F_i Then $F_i(d_p) > F_i(d_q)$

2.16.5 Genetic Operator

Kode GA menggunakan *Simulated Binary Crossover* (SBX). Untuk genetic operator ini terdiri dari *Simulated Binary Crossover* dan *Polynomial Mutation*.

2.16.6 Recombination dan Seleksi

Populasi *Offspring* digabungkan dengan generasi populasi yang terbaru. Seleksi membentuk kumpulan individu untuk generasi selanjutnya dan dipilih yang terbaik untuk ditambahkan pada populasi. Populasi dikelompokkan berdasarkan non dominasi. Generasi baru diisi oleh masing – masing *front* hingga ukuran populasi melampaui ukuran populasi baru. Jika penambahan semua individu di *front* F_j populasinya melebihi N maka dinividu di F_j dipilih berdasarkan *crowding distancenya* secara menurun sampai ukuran populasinya N dan prosesnya diulang untuk membangkitkan generasi berikutnya.

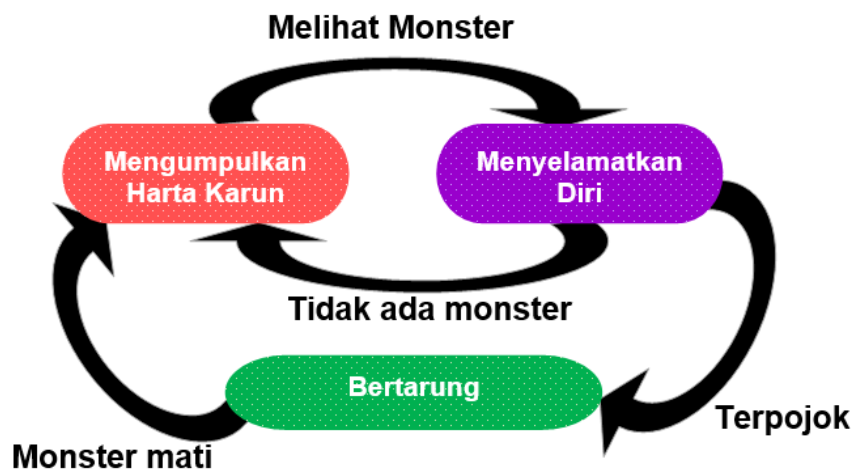
2.18 Finite State Machine (FSM)

Finite State Machine (FSM) atau *Finite State Automata* (FSA) [27] merupakan metode yang paling populer digunakan pada *game*. Metode ini seakan menjadi standard an telah digunakan secara luas dibidang *game*. Hal ini karena kesederhanaan dan kemudahan pengimplementasian FSM pada *game*.

FSM pada adalah pemodelan dari perilaku sebuah system/objek dengan beberapa kondisi yang terdefiniskan. Di mana objek tersebut dapat bertransisi ke *state* tertentu setiap waktu dengan batasan angka dari kondisi atau objek yang ditentukan. FSM terdiri dari dua elemen utama yaitu *state* (keadaan) dan *transition* (transisi). *State* merupakan keadaan objek saat ini, sedangkan *transition* merupakan hal yang dilakukan agar bisa berpindah dari satu *state* ke *state* yang lain. *Transition* juga bisa menyatakan kondisi yang harus dipenuhi oleh objek agar bisa berpindah dari satu *state* ke *state* yang lain. Sedangkan menurut [35], FSM terdiri dari 4 elemen utama yaitu :

1. *State* yang mendefinisikan perilaku dan mungkin menghasilkan aksi
2. Transisi *state* merupakan pergerakan dari satu *state* ke *state* yang lainnya
3. Aturan atau kondisi dimana harus terpenuhi supaya ada transisi *state*
4. Kejadian (*event*) input yang terjadi baik internal maupun eksternal, dimana kemungkinan dipicu oleh aturan dan mengacu ke transisi *state*.

FSM digambarkan sebagai jaringan semantic yang merepresentasikan maksud dan hubungan dengan menggunakan kata – kata. *State* dilambangkan dengan lingkaran. Sedangkan *transition* disimbolkan dengan anak panah dengan arah tertentu. Tiap Lingkaran dan anak panah memiliki nama masing – masing yang menunjukkan status mereka. Contoh sederhana dari FSM pada Gambar 2.23.



Gambar 2.13 Contoh FSM

Sumber : Jared Rain, FSM in Game, 2004

State yang ada adalah mengumpulkan harta karun, menyelamatkan diri dan bertarung. Sedangkan transisi adalah melihat monster, tidak melihat monster, terpojok, dan monster mati. Jadi dapat dikatakan jika objek tersebut sedang berada di *state* “mengumpulkan harta karun” lalu melihat monster maka dia akan berpindah ke *state* “menyelamatkan diri”. Demikian seterusnya.

Halaman ini Sengaja Dikosongkan

BAB 3

METODOLOGI PENELITIAN

Pada *Clash Of Clans*, Barbarian King dan Archer Queen adalah 2 contoh pahlawan yang sama – sama memiliki satu tujuan yaitu menghancurkan bangunan musuh sebanyak – banyaknya, tidak ada target khusus, perbedaan hanya terdapat pada jarak target serangannya. Barbarian King menyerang bangunan musuh pada jarak dekat karena senjatanya adalah pedang, Archer Queen menyerang bangunan musuh pada jarak jauh karena senjatanya adalah panah. Saat di *deploy* mereka hanya fokus pada target serangannya tanpa melihat sedikitpun kondisi pasukan yang ia miliki pada saat pertarungan berlangsung, hal ini selain membuat mereka cepat mati, juga membuat peran mereka sebagai agenp terkuat menjadi sia – sia dan membosankan karena perilaku mereka sama dengan pasukan lainnya (barbarian, archer, giant, goblin, wizard, hog rider, valkry, pekka, healer). Keahlian *support* atau *ability* hanya dimiliki oleh Barba

rian King, dimana *ability* nya hanya memanggil pasukan barbarian yang berjumlah 5 untuk membantunya menghancurkan target yang ia serang. Pahlawan dalam game harus memiliki perilaku yang berbeda dengan pasukan yang berada di bawahnya, tidak hanya dapat menyerang, tapi juga memiliki keahlian memimpin, men-*support*, dan pahlawan yang juga memiliki sebuah keterbatasan, keterbatasan pahlawan yang diambil menjadi penelitian adalah egois. Egois yang terbagi menjadi dua, yaitu egois yang cenderung melindungi diri sendiri tanpa melihat keadaan orang lain (pasukannya) dan egois yang masih melihat keadaan orang lain (pasukannya) sehingga ia akan melarikan diri dengan mengajak pasukannya tersebut. Perilaku *natural* pahlawan dalam dunia nyata maupun dalam film inilah yang diharapkan dapat menghidupkan tantangan dan kemenarikan permainan, yang membuat pahlawan dalam *game* menjadi unik.

Bab ini akan menjelaskan tahapan-tahapan yang dilakukan pada penelitian untuk mencapai tujuan tersebut. Dan hasil dari penelitian ini akan diterapkan pada

Batara, Wira, dan Kirna untuk menentukan perilaku yang optimal ketika dihadapkan dengan berbagai kemungkinan kondisi musuh yang ditemui.

3.1 Lingkup Penelitian

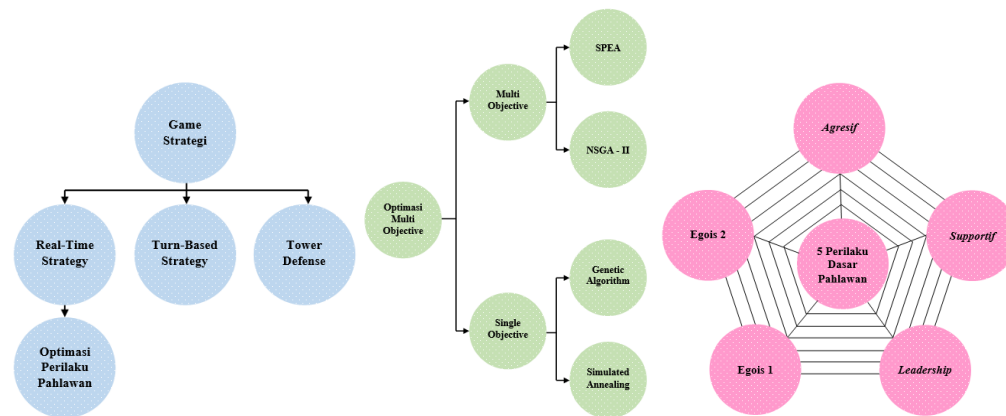
Tujuan penelitian ini adalah mengoptimasi perilaku pahlawan yang multiobjective dimana lima perilaku pahlawan mengacu pada lima sifat dasar kepribadian yang telah dilakukan oleh beberapa psikolog yaitu *extraversion* membuat pahlawan berperilaku *agresif*, *conscientiousness* yang membuat pahlawan berperilaku *leadership*, *neuroticism* yang membuat pahlawan berperilaku egois 1 dan egois 2, seperti *agreeableness* yang membuat pahlawan berperilaku *supportif*. Sehingga pahlawan dalam game tidak hanya memiliki *single objective* tapi *multiobjective*, bukan hanya menjadi agen terkuat namun pahlawan memiliki kepribadian yang membuatnya berbeda dengan pasukannya. Pemberian ruang lingkup penelitian ditujukan agar penelitian lebih terarah. Penelitian ini merupakan irisan dari tiga topik yaitu Strategy Game, Dynamic Difficulty Adjustment, dan Optimization Method. Diagram Venn mengenai lingkup penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Venn Lingkup Penelitian

Berdasarkan ketiga topik penelitian yang diambil, dikerucutkan menjadi suatu bagian spesifik dari topik-topik tersebut dikarenakan besarnya ruang lingkup dari topik penelitian. Topik Strategy game yang menjadi permasalahan pada penelitian ini, diambil sub-genre *Real Time Strategy* untuk dijadikan bahan penelitian yang nantinya akan lebih difokuskan kepada optimasi perilaku pahlawan.

Lima perilaku dasar pahlawan yang merupakan asumsi dari penelitian para psikolog yaitu Lima sifat dasar kepribadian manusia, yang akan dijelaskan secara rinci pada bab setelahnya. Serta NSGA-II yang digunakan sebagai metode optimasi yang mana merupakan metode optimasi dengan multi-obyektif. Berikut pohon penelitian yang ditunjukkan oleh Gambar 3.2 :



Gambar 3.2 Pohon Penelitian

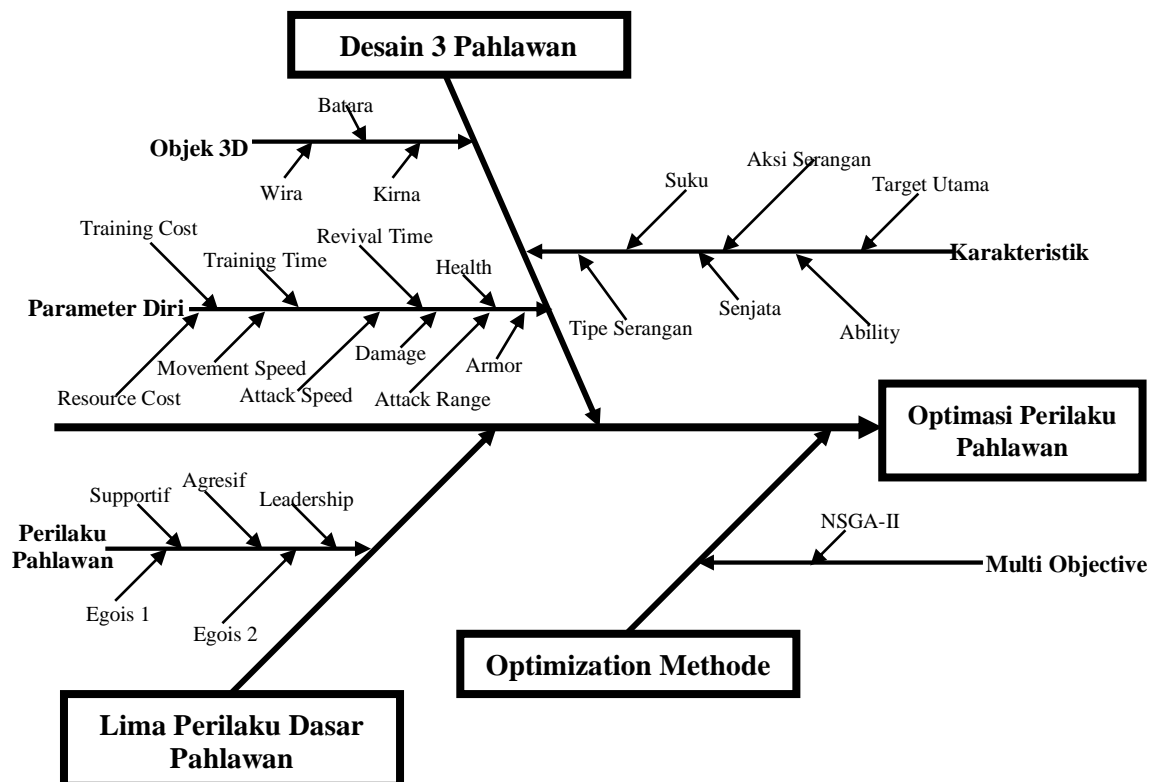
3.2 Deskripsi Permainan

Game DWIPA YUDHA merupakan sebuah game bergenre RTS (*Real Time Strategi*) dimana latar belakang dari game ini adalah sebuah kerajaan besar di Indonesia yang telah dipimpin oleh seorang Raja yang memiliki 3 orang putra. Ketiga putra nya ini tidak pernah akur satu sama lain. Raja yang sadar bahwa semakin hari semakin menua dan ia mulai tak sanggup lagi memimpin kerajaannya ini, beliau mulai berfikir untuk mewariskan tahtanya kepada salah satu dari ketiga putranya tersebut. Beliau akhirnya menemukan cara agar ia dapat menemukan siapa dari salah satu putranya yang pantas menggantikan tahtanya, yaitu dengan memberikan mereka 3 wilayah kerajaan yang berbeda. Ketiga wilayah tersebut adalah Madura, Jawa, dan Bali. Dengan memberikan mereka 3 wilayah yang berbeda, ia dapat memantau dan mengetahui wilayah mana yang lebih unggul yaitu salah satu putranya yang pantas menggantikannya menjadi raja. Sehingga pertempuran antar ketiga wilayah inilah yang mengawali perebutan tahta raja . Ketiga putra tersebut memiliki teman semasa kecil yang nantinya juga akan mengikuti mereka dalam memimpin setiap wilayah tersebut. Dan teman masa kecil mereka adalah pahlawan yang berada dalam penelitian ini. Pahlawan tersebut lah

yang akan berperan penting menjaga wilayah tersebut, sehingga berdasarkan latar belakang ini pahlawan benar – benar harus menjalankan tugasnya dengan baik, pahlawan harus berperilaku selayaknya seorang pahlawan yang tak hanya memiliki jiwa berani mati untuk melawan musuhnya, namun juga harus dapat melindungi yang lemah yaitu pasukannya.

3.3 Tahapan Pembuatan Sistem

Dalam permainan DWIPA YUDHA, penelitian difokuskan pada optimasi perilaku pahlawan. Untuk mengoptimasi perilaku pahlawan dalam, harus melewati berbagai langkah atau tahapan, mulai dari mendesain ketiga pahlawan tersebut berupa object 3 Dimensi yang nantinya akan digunakan untuk simulasi, kemudian desain karakter ketiga pahlawan tersebut sesuai suku atau *clan* nya masing – masing, karena tujuan penelitian ini adalah optimasi perilaku pahlawan yang multi objective maka diperlukan desain fungsi objective untuk masing – masing tujuan pahlawan tersebut, penggunaan metode untuk mengoptimasi, hingga simulasi nya yang akan ditunjukkan dengan *game engine* Unity.



Gambar 3.3 Diagram Fishbone Optimasi Perilaku Pahlawan

3.4 Desain 3 Pahlawan DWIPA YUDHA

Bila pada game *Clash Of Clans* memiliki 2 pahlawan yaitu Barbarian King dan Archer Queen, maka dalam game DWIPA YUDHA yang sangat ingin mengangkat budaya kental Indonesia, memiliki 3 pahlawan yaitu Batara Maheswara, Wira Oragastra, dan Kirna Waranggani yang masing – masing mewakili beberapa suku yang terkenal dengan ciri khas nya di Indonesia, yaitu Suku Madura, Suku Jawa, dan Suku Bali.

3.4.1 Desain Objek 3 Dimensi (3D) Pahlawan

Dimana desain setiap pahlawan tersebut yaitu Pakaian dan Senjata disesuaikan dengan *Clan* atau Suku masing – masing. Batara Maheswara yang berasal dari Suku Madura memiliki pakaian yang identik dengan garis – garis merah putih dan senjata clurit sesuai dengan ciri khas mayoritas Suku Madura. Wira Oragastra yang berasal dari Suku Jawa memiliki pakaian yang identik dengan batik dan senjata panah sesuai dengan ciri khas mayoritas Suku Jawa. Begitupun dengan Kirna Waranggani, satu – satunya pahlawan wanita, ia berasal dari Suku Bali memiliki pakaian yang identik dengan motif kotak – kotak hitam putih dan senjata sihir sesuai dengan ciri khas mayoritas Suku Bali. Gambar 3.2 dibawah ini akan menampilkan desain setiap pahlawan dalam DWIPA YUDHA :



Gambar 3.4 Desain 3D pahlawan DWIPA YUDHA

(a) Batara Maheswara (b) Wira Oragastra (c) Kirna Waranggani

Ketiga pahlawan ini memiliki arti disetiap namanya, berikut penjelasannya :

1. **Batara Maheswara** : Batara = Kewibawaan Raja; Maheswara = Raja Besar
Maka artinya, Kewibawaan Raja yang Besar
2. **Wira Oragastra** : Wira = Pahlawan; Oragastra = Anak panah
Maka artinya, Pahlawan yang membawa Anak Panah
3. **Kirna Waranggani** : Kirna = Sinar yang cantik dan molek; Waranggani = berwajah elok, maka artinya : Wajah elok yang memiliki sinar cantik dan molek

3.4.2 Desain Karakteristik Pahlawan

Setiap pahlawan memiliki karakteristik masing – masing sesuai dengan suku nya. Suku Madura yang terkenal dengan keberaniannya, Suku Jawa yang terkenal dengan kesederhanaannya, serta Suku Bali yang terkenal dengan keramahannya, semua masuk dalam diri setiap pahlawan dan dikemas berupa tabel pada tabel 3.1. Tabel tersebut akan menampilkan rincian setiap parameter yang dimiliki setiap pahlawan mulai dari target utama serangan hingga keahlian *support* atau *ability* yang dimiliki masing – masing pahlawan. Karakteristik yang pertama adalah perbedaan target serangan, dimana Suku Madura yang terkenal dengan keberaniannya, memiliki target serangan yang memiliki resiko tinggi yaitu Tower Defense musuh yang terdiri dari Gandewa tower, Gayatri Tower, Adikara, dan Kamandaka. Sedangkan pahlawan Suku Jawa yang terkenal dengan kesederhanaannya lebih memilih target serangan yang tidak terlalu beresiko, yaitu *Resource* musuh atau yang lebih dikenal dengan harta kekayaan musuh yang terdiri dari emas, batu, dan bahan makanan. Serta pahlawan Suku Bali yang terkenal dengan keramahannya tidak memiliki target khusus, apapun bangunan yang berada didekatnya akan ia serang. Aksi serangan tiap pahlawan berbeda menurut senjata yang dibawa, bila Batara dari Suku Madura yang memiliki senjata khas clurit, maka aksi serangannya adalah gerakan – gerakan seseorang yang memegang sebuah clurit yang menyerang targetnya dari jarak dekat, yaitu Swing, dan Slash. Swing adalah gerakan memegang clurit dengan kedua tangan dan kekuatan penuh sedangkan Slash adalah gerakan memegang clurit dengan satu tangan dan kekuatan sedang. Wira dari Suku Jawa yang memiliki senjata khas panah yang menyerang targetnya dari jarak jauh, aksi serangannya adalah gerakan seseorang yang memegang sebuah

panah dengan busurnya, gerakannya sama hanya anak panahnya yang dikeluarkannya berbeda kekuatan, bila kekuatan penuh mengeluarkan anak panah berupa api, maka dengan kekuatan yang biasa akan mengeluarkan anak panah biasa. Begitupun dengan Kirna dari Suku Bali yang memiliki senjata khas berupa sihir yang menyerang targetnya dari jarak dekat maupun jarak jauh, maka gerakannya adalah gerakan seseorang yang sedang mengeluarkan sihirnya. Yang berbeda pada kekuatannya, bila pada kekuatan penuh ia mengeluarkan *fire ball* maka dengan kekuatan sedang ia mengeluarkan *lightning storm*. Untuk Keahlian Support masing – masing pahlawan, dimulai dengan Batara yang berasal dari Suku Madura, dengan kekuatannya yang lebih besar atau lebih kuat dari Wira dan Kirna, ia memiliki keahlian *support* atau *ability* pada kepemimpinannya (*leadership*) dengan menambah kesehatan pasukannya. Dimana bila terdapat pasukan di sekitar wilayah *range* Batara dan kondisi mereka lemah, dikatakan lemah adalah ketika terdapat 2 tipe pasukan (misal : sakera dan gandewa) yang memiliki level terendah yaitu level 1 dan jumlah pasukannya tersebut sedikit (1 hingga 3), maka Batara akan otomatis mengeluarkan *ability* nya tersebut. Wira yang kekuatannya lebih rendah dari Batara memiliki keahlian *support* atau *ability* menambah kecepatan serangan pasukannya, sama dengan Batara, Wira menambah kecepatan serangan pasukannya ketika ia memilih perilaku *Supportif*, dimana bila terdapat pasukan di sekitar wilayah *range* Wira dan kondisi mereka lemah, dikatakan lemah adalah ketika terdapat 2 tipe pasukan (misal : sakera dan gandewa) yang memiliki level terendah yaitu level 1 dan jumlah pasukannya tersebut sedikit (1 hingga 3). Sedangkan Kirna yang kesehatan dan *damagenya* berada di tengah – tengah Batara dan Wira, dimana kesehatannya lebih rendah dari Batara dan Wira namun *damage* nya lebih tinggi dari Wira dan lebih rendah dari Batara, maka ia memiliki keahlian *support* atau *ability* menambah *damage* yang dihasilkan oleh pasukannya. Sama dengan Batara dan Wira menambah kecepatan serangan pasukannya ketika ia memilih perilaku *Supportif*, dimana bila terdapat pasukan di sekitar wilayah *range* Wira dan kondisi mereka lemah, dikatakan lemah adalah ketika terdapat 2 tipe pasukan (misal : gandewa dan gayatri) yang memiliki level terendah yaitu level 1 dan jumlah pasukannya tersebut sedikit (1 hingga 3).

Tabel. 3.1 Karakteristik Pahlawan DWIPA YUDHA

Pahlawan	Suku	Target Utama	Aksi Serangan	Senjata	Tipe Serangan	Ability
Batara Maheswara	Rajapajan (Madura)	Bangunan Pertahanan Musuh	Swing Slash	Clurit / Pedang	Jarak Dekat	Menambah Kesehatan Pasukan
Wira Oragatra	Bronggalan (Jawa)	Bangunan Sumber daya musuh (resource)	Panah Api Panah Biasa	Panah	Jarak Jauh	Menambah Kecepatan Serangan Pasukan
Kirna Waranggani	Sidotopo (Bali)	Bangunan Terdekat	Fire Ball Lightning Storm	Sihir	Jarak Dekat dan Jauh	Menambah efek serangan (damage) Pasukan

3.4.3 Parameter Diri Setiap Pahlawan

Pada Tabel 3.2, 3.3, 3.4 berisi parameter setiap pahlawan mulai dari kesehatan yang dimiliki, hingga waktu pemulihan kesehatan pahlawan untuk level 1 sampai level maksimal yaitu level 3. Berikut penjelasan dari setiap parameter pahlawan :

- 1. Damage** : Kerusakan yang dihasilkan kibasan senjata per detik (%)
- 2. Armor** : Ketahanan yang dimiliki pahlawan
- 3. Health** : Kesehatan yang dimiliki pahlawan
- 4. Attack Speed** : Kecepatan serangan setiap aksi atau gerakan senjata per detik
- 5. Movement Speed**: Kecepatan gerakan berjalan atau berpindahnya pahlawan per detik
- 6. Training Cost** : Harga yang dikeluarkan untuk melatih 1 pahlawan
- 7. Training Time** : Waktu yang diperlukan untuk melatih 1 pahlawan
- 8. Research Cost** : Harga pada laboratorium untuk setiap peningkatan level pahlawan
- 9. Research Time** : Waktu yang diperlukan laboratorium untuk peningkatan level pahlawan
- 10. Attack Range** : Wilayah atau *range* kekuasaan serangan pahlawan

- 11. Revival Time** : Waktu pemulihan kesehatan pahlawan setelah menyelesaikan Pertempuran

Tabel. 3.2 Rincian Parameter Batara Maheswara

RAJAPAJAN HERO : BATARA MAHESWARA SPELIALIS : CLURIT			
Status	Level 1	Level 2	Level 3
Damage	20	30	50
Armor	5	5	10
Health	100	120	150
Attack Speed	1/s	2/s	3/s
Movement Speed	1	2	3
Training Cost	-	100.000	1.000.000
Training Time	200s	-	250s
Research Cost	-	5*Sakera	7*Sakera
Research Time	-	500 s	510 s
Attack Range	1	1	1
Revival Time	250	350	450

Tabel. 3.3 Rincian Parameter Wira Oragastra

BRONGGALAN HERO : WIRA ORAGASTRA SPELIALIS : PANAHA			
Status	Level 1	Level 2	Level 3
Damage	7	13	20
Armor	4	5	7
Health	50	70	100
Attack Speed	2/s	2/s	3/s
Movement Speed	2	2	4
Training Cost	-	100.000	1.000.000
Training Time	40 s	45 s	50 s
Research Cost		5*Gandewa	7*Gandewa
Research Time	-	500 s	510 s
Attack Range	6	6	7
Revival Time	250	350	450

Tabel. 3.4 Rincian Parameter Kirna Waranggani

SIDOTOPO HERO : KIRNA WARANGGANI SPESIALIS : SIHIR			
Status	Level 1	Level 2	Level 3
Damage	8	12	25
Armor	5	6	8
Health	50	60	80
Magical Point	220	260	300
Attack Speed	1/s	1/s	2/s
Movement Speed	2	3	3
Training Cost	-	100.000	1.000.000
Training Time	20 s	30 s	40 s
Research Cost	-	5*Gayatri	7*Gayatri
Research Time	-	215 s	240 s
Attack&Heal Range	6	8	8
Heal Speed/Mana cost	5/12	8/10	8/10
Regen Mana	1/s	1.5/s	2/s
Revival Time	250	350	450

3.5 Lima Perilaku Pahlawan

Seorang pahlawan dalam sebuah game pada umumnya hanya memiliki satu tujuan yaitu memenangkan pertempuran. Pahlawan dalam game Clash Of Clans (Barbarian King dan Archer Queen) dan game lain yaitu Song of Runiation 2 [29] hanya single objective, dimana mereka hanya dapat menyerang hingga nyawa atau kesehatan mereka habis. Mereka berperang secara individual, hanya berfokus untuk menghancurkan atau membunuh musuh sehingga dapat memenangkan pertempuran. Padahal dengan perilaku mereka yang hanya bisa menyerang, dapat membuat mereka cepat mati sehingga permainan ini menjadi membosankan. Peran penting seorang pahlawan menjadi tidak terlihat seperti pahlawan yang kita tahu pada dunia nyata maupun di film, pahlawan yang seharusnya adalah pahlawan yang tidak hanya rela mati namun juga dapat melindungi sesamanya terutama yang lemah, dapat menjadi pemimpin, dan tak hanya memiliki semua kelebihan, pahlawan adalah manusia biasa yang memiliki keterbatasan pula yaitu keegoisan. Agar permainan menjadi menarik, menantang, dan membuat penasaran, seorang

pahlawan yang memiliki kepribadian yang mengacu pada penelitian psikolog yaitu *Big Five Personality*, akan menjadi kesan tersendiri bagi pemain *game* ini. Bisa saja yang awalnya pemain hampir kalah karena nyawa pasukan hampir habis, dengan adanya *objective* pahlawan yang dapat melindungi pasukannya yaitu berupa *support* kesehatan, *damage*, maupun kecepatan serangan, pasukannya akan dapat bertahan dan dapat terus menyerang, begitupun dengan jiwa kepemimpinan yang dimiliki pahlawan, dengan kepemimpinannya tersebut pahlawan dan pasukan yang awalnya terancam mati, akhirnya dapat tetap menyerang bersama – sama sehingga kekalahan pemain dapat berubah menjadi kemenangan. Pahlawan memiliki keterbatasan layaknya manusia biasa, terkadang bila ia dihadapkan oleh suatu keadaan yang sangat tidak memungkinkan untuk bertahan hidup ia memiliki sebuah keegoisan dengan cara melarikan diri tanpa melihat kondisi sekelilingnya. Bila dalam *game* sebuah NPC hanya diberi kecerdasan buatan atau *Artificial Intelligent* maka dari penelitian ini, pahlawan memiliki keterbatasan yang memperlihatkan bahwa ia bisa saja mementingkan dirinya sendiri setelah melihat musuhnya yang lebih kuat dari padanya dan pasti ia akan langsung mati bila langsung menyerang. Sehingga dengan adanya perilaku – perilaku yang merupakan gabungan dari *Artificial Intelligent* dan keterbatasan yang dimiliki ini, pahlawan dalam *game* akan menjadi sangat *uniq*, tidak mudah ditebak perilakunya apabila dihadapkan dengan berbagai macam kondisi musuhnya. Pahlawan dalam *game* DWIPA YUDHA ini memiliki kelima perilaku tersebut, rela mati dengan menghancurkan seluruh bangunan musuhnya (*Agresif*), dapat menjadi seorang pendukung dan pelindung bagi pasukannya (*Supportif*), menjadi pemimpin dalam pertempuran (*Leadership*) serta memiliki keterbatasan yaitu Keegoisan, dimana keegoisannya ini terbagi menjadi 2 yaitu : Egois 1 yang mementingkan dirinya sendiri dengan cara melarikan diri tanpa melihat kondisi pasukannya, dan Egois 2 yang mementingkan dirinya dan pasukannya dengan cara mengajak pasukannya tersebut melarikan diri pula. Berikut penjelasan kelima perilaku pahlawan DWIPA YUDHA :

1. **AGRESIF** : sebuah perilaku dari pahlawan dimana perilakunya ini adalah terus menerus menyerang demi menghancurkan bangunan musuh sebanyak – banyaknya ataupun menguras habis harta kekayaan musuhnya. Pahlawan

ini akan berperilaku *agresif* bila kesehatannya atau nyawanya lebih besar daripada musuhnya dan *damage* yang diterimanya lebih kecil dari *damage* yang diterima musuhnya.

2. **SUPPORTIF** : sebuah perilaku dari pahlawan dimana perilakunya ini adalah melindungi dan mendukung yang lemah darinya, yaitu pasukannya. Perlindungan atau dukungan yang dapat dilakukan pahlawan untuk pasukannya yaitu dengan memberikan *support* baik berupa tambahan kesehatan, tambahan *damage*, serta tambahan kecepatan serangan. Tergantung karakteristik masing – masing pahlawan. Dengan adanya *support* dari pahlawan ini, maka pasukan yang awalnya lemah atau nyawanya hampir habis tetap dapat menyerang musuhnya. Pahlawan ini akan berperilaku *supportif* bila akumulasi nyawa pasukannya lebih rendah daripadanya dan akumulasi *damage* yang diterima pasukannya lebih besar daripada *damage* yang diterima musuhnya.
3. **LEADERSHIP** : sebuah perilaku dari pahlawan dimana perilakunya ini adalah memimpin pasukannya yang memiliki akumulasi nyawa lebih tinggi daripada nyawanya. Pahlawan akan memimpin penyerangan dimana pasukan yang berada dalam *range* wilayahnya diperintahkan untuk menyerang bersama – sama dengannya target serangan yang diserang olehnya, sehingga ancaman untuknya karena nyawanya yang lebih kecil, akan berubah jadi penyatuan kekuatan bersama untuk menghancurkan musuh – musuhnya. Pahlawan ini akan berperilaku *leadership* bila akumulasi nyawa pasukannya lebih tinggi daripadanya dan akumulasi *damage* yang diterima pasukannya lebih kecil daripada *damage* yang diterima musuhnya.
4. **EGOIS 1** : sebuah perilaku dari pahlawan dimana perilakunya ini adalah mengarah kepada kebodohan atau keterbatasan seperti yang dimiliki manusia biasa karena yang kita ketahui pahlawan tetaplah seorang manusia biasa. Kebodohan atau keterbatasannya ini adalah sebuah keegoisan melarikan diri karena ia dihadapkan oleh keadaan dimana nyawa yang dimilikinya lebih

kecil dari nyawa musuhnya, dan *damage* yang diterimanya juga lebih besar daripada nyawa musuhnya. Sehingga, ketika pertama kali ia dihadapkan pada sebuah pertempuran, yang ada adalah ia akan langsung mati.

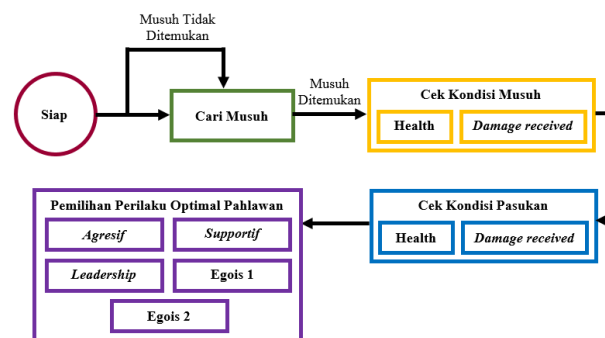
5. **EGOIS 2** : sebuah perilaku yang mengarah pada suatu kebodohan atau keterbatasan lainnya, namun perbedaannya dengan adalah, pada egois 1 ia hanya melihat kondisinya tanpa melihat kondisi pasukannya, namun pada egois 2 ini ia masih melihat kondisi pasukannya. Egois 2 ini adalah perilaku dimana pahlawan melarikan diri namun ia juga mengajak pasukannya melarikan diri, karena ia melihat kondisi pasukannya berada dalam kondisi seperti dirinya, nyawa pasukannya yang kecil dan *damage* yang diterima pasukannya lebih besar dari musuhnya.

3.6 Desain Skenario Pahlawan dan pasukan terhadap Musuh

Hasil dari penelitian ini dapat diterapkan untuk perilaku Batara Maheswara, Wira Oragastra, dan Kirna Waranggani atau Pahlawan lain yang membutuhkan optimasi perilaku untuk beberapa tujuan atau Multi Objective Optimization. Pada awalnya akan dicoba perbandingan dua tujuan atau dua objective yang saling bertentangan, dimana menampilkan pahlawan yang tetap hidup namun pasukannya mati (*agresif*) dan pahlawan yang mati namun pasukannya tetap hidup (*supportif*). Dengan optimasi, dua perilaku yang bertentangan tersebut tak perlu mengorbankan nyawa siapapun, baik nyawa pahlawan maupun nyawa musuhnya. Dengan optimasi, kedua tujuan yang saling bertentangan akan dapat dicapai, yaitu menghancurkan musuh dengan pahlawan yang masih tetap hidup dan melindungi pasukan. Bila kedua objective yang saling bertentangan tersebut terpenuhi, maka dapat dilanjutkan dengan mengoptimalkan lima objective saling bertentangan. ermodelan mengenai siklus hidup (*life cycle*) dari pahlawan dalam *game* DWIPA YUDHA menggunakan FSM.

Pada penelitian ini, FSM menunjukkan perilaku Pahlawan yang terdiri dari beberapa state (kondisi) dan transition (perpindahan) seperti yang ditunjukkan pada Gambar 3.4. Berdasarkan FSM pahlawan tersebut ditunjukkan bahwa pahlawan memulai dari kondisi siap di-*deploy* untuk bertempur, pahlawan yang siap

bertempur adalah pahlwan yang sudah cukup asupan makanannya sehingga kesehatannya penuh. Kemudian pahlwan membangkitkan gen awal secara acak yang memiliki rentang dari 0 hingga 1, sehingga kondisi siap di-*deploy* berpindah ke kondisi melihat kondisi kesehatannya, setelah mengetahui kondisi kesehatannya maka yang pahlwan lakukan selanjutnya adalah melihat kondisi kesehatan musuhnya, setelah ia mengetahui kondisi kesehatan musuhnya, pahlwan mulai membandingkan kondisi kesehatannya terhadap musuhnya. Setelah ia mengetahui kondisi kesehatan musuhnya, yang pahlwan lakukan selanjutnya adalah melihat atau melihat *damage* yang akan diterimanya, setelah melihat *damage* yang diterimanya, ia melihat *damage* yang diterima musuhnya. Setelah mengetahui dua kondisi tersebut, ia mulai melihat kondisi kesehatan pasukannya. Pahlwan membandingkan pula kesehatannya terhadap pasukannya, dan membandingkan kesehatan pasukannya terhadap musuhnya. Begitupun dengan *damage* yang akan diterima pasukannya, ia melihat kondisi tersebut hingga akhirnya perilaku yang optimal ketika dihadapkan oleh musuh yang seperti itu dengan kondisi pasukan yang seperti itu pula keluar.



Gambar 3.5 FSM Pahlwan dalam permainan

Pada penelitian ini dibutuhkan perilaku yang optimal dari seorang pahlwan untuk dapat memenangkan pertempuran ketika dihadapkan pada lima tujuan yang bertentangan yaitu antara menghancurkan seluruh bangunan musuh beserta menguras habis harta musuh, melindungi dan mendukung pasukannya, memimpin pertempuran, melindungi atau menjaga nyawanya sendiri dan menjaga atau melindungi nyawa pasukannya. Bila pahlwan menghancurkan seluruh bangunan musuh beserta menguras habis harta musuh maka ia akan mengarahkan

semua kekuatannya untuk mencapai tujuannya tersebut sehingga pahlawan tersebut berperilaku *agresif*. Seorang pahlawan juga harus menjadi pelindung atau pendukung untuk orang lain yang lemah dalam hal ini adalah pasukannya. Maka untuk mencapai tujuannya itu ia akan mengeluarkan *ability*nya mulai dari menambah kesehatan, menambah *damage*, atau menambah kecepatan serangan pasukan – pasukan yang berada dalam *range* wilayah kekuasaannya. Selain itu pahlawan juga harus dapat memimpin, ketika keadaan kesehatannya tidak memungkinkan untuk menghadapi musuhnya yang lebih tinggi daripadanya, dengan melihat kondisi pasukannya yang lebih unggul daripadanya, dalam hal ini unggul adalah dari segi nyawa yang lebih banyak (akumulasi dengan jumlah pasukan yang ada) dan *damage* yang diterima pasukannya juga lebih kecil dari *damage* yang diterima pahlawan itu sendiri. Dengan kondisi pasukan yang lebih unggul ini, pahlawan berperan penting dalam memimpin pasukannya tersebut sehingga meskipun keadaannya tidak memungkinkan untuk menyerang musuh, dengan kepemimpinannya tersebut ia membuat pasukannya dapat meng-*cover*nya sehingga yang awalnya ia tak bisa menyerang, menjadi bisa menyerang dan itu bersama – sama dengan pasukannya. Namun di satu sisi pahlawan memiliki kondisi sangat lemah, dimana ia tidak akan mungkin dapat menyerang musuhnya, dan pasti ia akan kalah. Maka dari kondisi keterbatasannya inilah ia akhirnya berperilaku egois, yaitu melarikan diri, tanpa melihat kondisi pasukannya disekitarnya, hanya dengan melihat kondisi pasukan yang jauh lebih tinggi daripadanya, maka ia akan langsung melarikan diri kembali ke base atau tempat aman. Keegoisan pahlawan tak hanya keegoisan dirinya sendiri, ada kalanya ia juga memiliki keegoisan yang sedikit memperdulikan orang lain. Keegoisan 2 inilah keterbatasan kedua yang dimiliki pahlawan, dimana dengan melihat kondisi pasukannya yang juga tidak memungkinkan untuk melawan musuh, sekalipun itu telah diakumulasi dengan jumlah pasukannya dan tetap lebih lemah dari musuhnya, maka ia akan mengajak pasukannya tersebut untuk melarikan diri pula dengannya. Dari kesemua perilaku, yang terdiri dari 5 perilaku yang mengacu pada penelitian para psikolog tentang *Big Five Personality* atau Lima besar sifat dasar untuk kepribadian ini yang akan membuat pahlawan dalam penelitian ini juga memiliki kepribadian layaknya pahlawan di dunia nyata maupun dalam film. Sehingga pahlawan dalam permainan

nantinya tidak hanya berstatus sebagai agen terkuat, namun perannya dalam dunia nyata atau dalam film yang biasa kita tahu adalah peran pahlawan yang sebenarnya.

Optimasi perilaku pahlawan ini didapat dari berbagai macam kemungkinan musuh yang ditemuinya. Terdapat 3 level untuk setiap pahlawan dimana Level 1 adalah level terendah pahlawan yaitu kondisi Lemah, Level 2 adalah level peningkatan kondisi pahlawan menjadi Sedang, serta Level 3 adalah level tertinggi pahlawan yaitu kondisi Kuat. Begitupun dengan kondisi pasukannya, pahlawan dalam operasi penyerangan nanti akan memiliki pasukan yang berada di bawahnya. Level 1 adalah level terendah pasukan yaitu kondisi Lemah, Level 2 adalah level peningkatan kondisi pasukan menjadi Sedang, serta Level 3 adalah level tertinggi pasukan yaitu kondisi Kuat. Mereka akan menyerang bersama – sama sehingga skenario disusun berdasarkan setiap kombinasi kondisi pahlawan dan kondisi pasukan dibandingkan dengan berbagai macam kondisi musuh yang akan ditemuinya. Dalam penelitian ini, yang digunakan menjadi parameter atau *decision variables* adalah *Hit point* (kesehatan) baik dari pahlawan itu sendiri, dari musuh, dan dari pasukannya, *Damage* (kerusakan) juga berasal baik dari pahlawan, musuh, dan pasukannya, dan *Armor* (ketahanan) yang hanya berasal dari pahlawan dan pasukan. Ketiga parameter ini menjadi penentu karena dalam pertempuran, yang mempengaruhi penyerangan adalah NPC atau agen sebuah game, dalam hal ini adalah Pahlawan, pasukan, dan Musuh yang berupa *tower defense* atau bangunan pelindung instalasi yang memiliki senjata untuk menyerang balik demi menjaga nyawanya.

Tabel - tabel dibawah ini akan menjelaskan *rule base* setiap kemungkinan kondisi musuh yang ditemui pahlawan itu bersama pasukannya berdasarkan status level masing - masing.

Keterangan Tabel :

Pahlawan → **B** : Batara Maheswara; **W** : Wira Oragastra; **K** : Kirna Waranggani
Pasukan → **Sak** : Sakera; **Gan** : Gandewa; **Gay** : Gayatri
Musuh → **GDT** : Gandewa Tower; **GYT** : Gayatri Tower; **Adi** : Adikara;
Kam : Kamandaka

Tabel 3.5 Tabel Kondisi Kesehatan Pahlawan dan Pasukan

STATUS	Batara	Wira	Kirna	Sakera	Gandewa	Gayatri
Lemah (Level 1)	100	50	50	40	40	40
Sedang (Level 2)	120	70	60	50	40	50
Kuat (Level 3)	150	100	80	60	50	60

Untuk menghitung kondisi kombinasi kesehatan, diperoleh bahwa Pahlawan setiap suku berjumlah 1 sehingga rumus minimal dan maksimal pencarian kesehatan pahlawan adalah :

$$\text{Min}_{\text{Health Hero}} = 1 * \text{health}_{\text{min Hero}} \quad (3.1)$$

$$\text{Max}_{\text{Health Hero}} = 1 * \text{health}_{\text{max Hero}} \quad (3.2)$$

Keterangan :

$\text{Min}_{\text{Health Hero}}$: Kesehatan Hero pada kondisi minimal

$\text{Max}_{\text{Health Hero}}$: Kesehatan Hero pada kondisi maximal

$\text{health}_{\text{min Hero}}$: Nilai kesehatan Hero pada kondisi minimal

$\text{health}_{\text{max Hero}}$: Nilai kesehatan Hero pada kondisi maximal

Sedangkan untuk jumlah pasukan, ditentukan oleh *rule base* untuk setiap Townhall. Level Townhall menentukan jumlah pasukan yang dapat dihasilkan. Maka rumus minimal dan maksimal pencarian kesehatan pasukan adalah :

$$\text{Min}_{\text{Health Troops}} = \text{jml}_{\text{min troops TH 1}} * \text{health}_{\text{min troops TH1}} \quad (3.3)$$

$$\text{Max}_{\text{Health Troops}} = \text{jml}_{\text{max troops TH 3}} * \text{health}_{\text{max troops TH3}} \quad (3.4)$$

Keterangan :

$\text{Min}_{\text{Health Troops}}$: minimal kesehatan pasukan yang dicari

$\text{Max}_{\text{Health Troops}}$: maximal kesehatan pasukan yang dicari

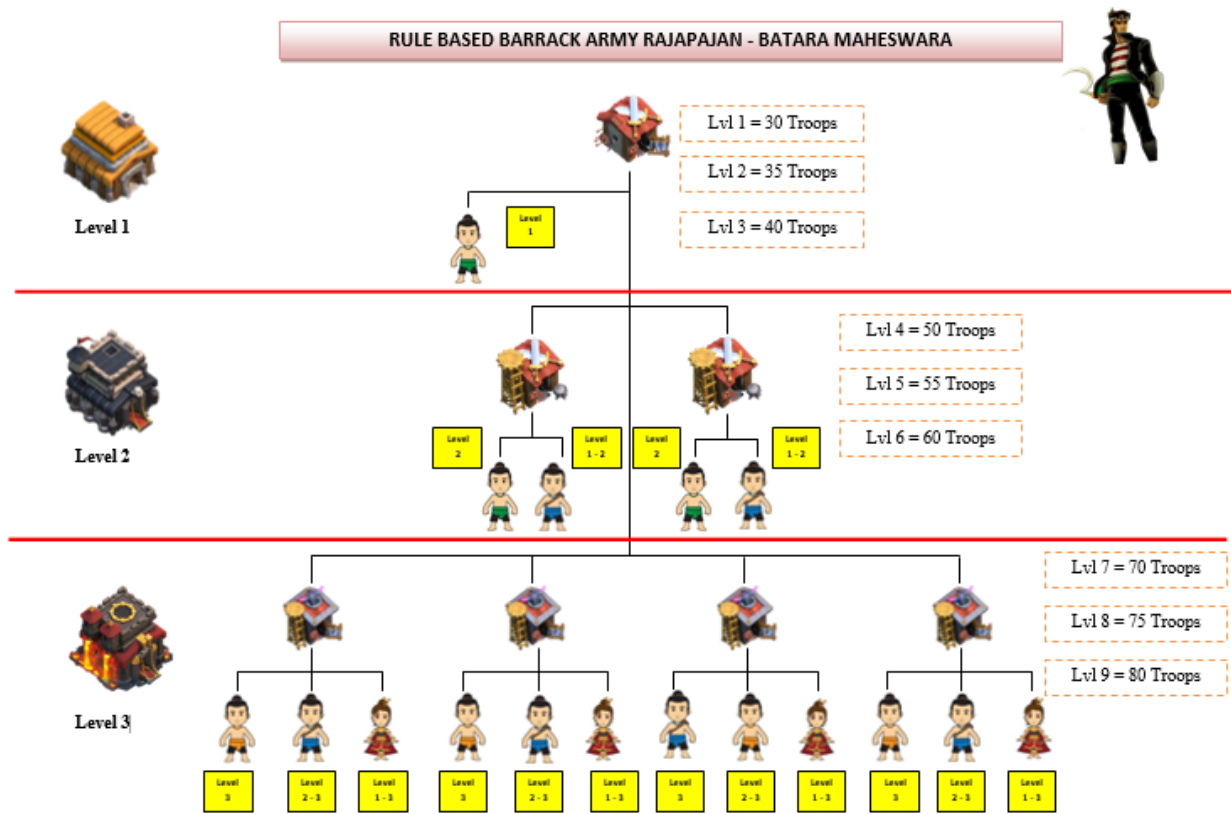
$\text{jml}_{\text{min troops TH 1}}$: jumlah minimal pasukan pada Townhall level 1

$\text{jml}_{\text{max troops TH 3}}$: jumlah maximal pasukan pada Townhall level 3

$\text{health}_{\text{min troops TH 1}}$: Nilai kesehatan minimal pasukan pada Townhall level 1

health_{max} troops TH 3 : Nilai kesehatan maximal pasukan pada Townhall level 3

Berikut *rule base* jumlah pasukan setiap Townhall :



Gambar 3.6 *Rule Base* Jumlah pasukan setiap Townha

Berdasarkan rumus (3.3) dan (3.4) diatas, maka seluruh perhitungan minimal dan maksimal ditunjukkan oleh Tabel berikut ini

Tabel 3.6 Tabel Kombinasi Kesehatan Pahlawan + Pasukan

Townhall 1	Batara	sakera	jumlah		Townhall 2	Batara	sakera + gandewa	jumlah	
min	100	1200	1300		min	120	2000	2120	
max	100	1600	1700		max	120	3000	3120	
Townhall 1	Wira	gandewa	jumlah		Townhall 2	Wira	sakera + gandewa	jumlah	
min	50	1200	1250		min	70	2000	2070	
max	50	1600	1650		max	70	3000	3070	
Townhall 1	Kirna	gayatri	jumlah		Townhall 2	Kirna	sakera + gayatri	jumlah	
min	50	1200	1250		min	100	2000	2100	
max	50	1600	1650		max	100	3000	3100	

Untuk *damage receive* pahlawan dan pasukan, merupakan *damage* yang diterima oleh pahlawan maupun pasukan. Dimana *damage received* ini merupakan hasil dari rumus pengurangan *damage* musuh dengan *damage* yang dihasilkan oleh pasukan maupun pahlawan tersebut :

$$\text{Damage Reduction} = ((0.06 * \text{Armor}) / (1 + 0.06 * \text{Armor})) * 100\% \quad (3.5)$$

$$\text{Total Damage} = \text{Enemy Damage} - \text{Damage Reduction} \quad (3.6)$$

Contoh Enemy Damage = 100 ; Armor = 5;

maka Damage Reduction = $((0.06 * 5) / (1 + 0.06 * 5)) * 100\% = 23,1\%$

Total Damage = $100 - (100 * 23,1\%) = 100 - 23,1 = 76,9$ *damage receive*

Keterangan Rumus :

Damage Reduction = *damage* atau kerusakan dihasilkan oleh pahlawan atau musuh

Armor = ketahanan yang dimiliki oleh pasukan atau musuh, ketahanan terhadap serangan musuh

Enemy Damage = *damage* atau kerusakan yang dihasilkan oleh musuh

Total Damage = Total kerusakan yang dihasilkan oleh pahlawan atau musuh

Maka berikut ini tabel setiap *damage received* yang diterima oleh pahlawan dan pasukan setelah dihitung berdasarkan rumus (3.4) dan (3.5).

Tabel 3.7 Tabel *Damage Receive* Pahlawan dan Pasukan

Batara		Damage Hero	Damage Pasukan
	Maksimal	280.5	288
	Minimal	3.4605	4.005
Wira		Damage Hero	Damage Pasukan
	Maksimal	288.5	294.9
	Minimal	3.168	4.005
Kirna		Damage Hero	Damage Pasukan
	Maksimal	285.6	298.7
	Minimal	3.4605	4.005

Untuk kesehatan musuh, didapat pula dari hasil akumulasi penjumlahan musuh setiap Townhall dengan kesehatannya, berikut kondisi jumlah musuh setiap Townhall (TH) beserta tabel perhitungannya :

A. GANDEWA TOWER

- TH Level 1 : 1
- TH Level 2 : -
- TH Level 3 : 1

2 →

Health : 325 – 575 [6 Level]
 Damage (/detik): 14 - 43
 Attack Range : 11
 Attack Speed : 0,5/s

B. GAYATRI TOWER

- TH Level 1 : -
- TH Level 2 : -
- TH Level 3 : 1

1 →

Health : 570 – 690 [4 Level]
Damage (/detik): 13 – 25
Attack Range : 8
Attack Speed : 1,2/s

C. ADIKARA

- TH Level 1 : -
- TH Level 2 : 2
- TH Level 3 : -

2 →

Health : 400 – 580 [4 Level]
Damage (/detik): 4,5 – 10,5
Attack Range : 15
Attack Speed : 4/s

D. KAMANDAKA

- TH Level 1 : 1
- TH Level 2 : 1
- TH Level 3 : 1

3 →

Health : 450 – 1030 [9 Level]
Damage (/detik): 10 – 62
Attack Range : 10
Attack Speed : 0,9/s

Dari kondisi diatas, maka akan dibuat tabel untuk menghitung setiap kesehatan musuh berdasarkan level Townhall nya. Karena setiap Townhall pada permainan RTS, terdapat *rule* atau aturan dimana Townhall level terendah atau terkecil, maka bangunan yang dapat dibangun hanya berjumlah 1 – 2 bangunan namun bangunan tersebut memiliki level pula yang dapat di-*upgrade* oleh pemain. Begitupun dengan Townhall level tertinggi, maka bangunan yang dapat dibangun lebih banyak dengan level maksimal yang dapat di-*upgrade*. Sebagai contoh, pada townhall 1 yang dapat dibangun hanya Gandewa Tower dan Kamandaka yang masing – masing berjumlah 1, namun dari masing – masing batas maksimal *upgrade* adalah level 2 dan 3. Sehingga minimal dan maksimal dari kesehatan musuh yang ditemui pahlawan ketika pertempuran adalah hasil perkalian dari jumlah bangunan dikalikan kesehatan dari level yang dimiliki bangunan tersebut. Untuk lebih jelasnya, akan ditunjukkan oleh Tabel 3.9 di bawah ini .

Tabel 3.8 Tabel Kesehatan Musuh

TH Level 1		Jumlah Bangunan	Level	Perhitungan	Hasil
	Gandewa Tower	1	1	HP Level 1 : 325 maka $1 \times 325 = 325$	
			2	HP Level 2 : 375 maka $1 \times 375 = 375$	325
	Gayatri Tower	-			
	Adikara	-			
	Kamandaka	1	1	HP Level 1 : 450 maka $1 \times 450 = 450$	

			2	HP Level 2 : 510 maka $1 \times 510 = 510$	
			3	HP Level 3 : 570 maka $1 \times 570 = 570$	450
				JUMLAH MINIMAL	775
TH Level 2		Jumlah Bangunan	Level		
	Gandewa Tower	1	1	HP Level 1 : 325 maka $1 \times 325 = 325$	
			2	HP Level 2 : 375 maka $1 \times 375 = 375$	
			3	HP Level 3 : 425 maka $1 \times 425 = 425$	
			4	HP Level 4 : 475 maka $1 \times 475 = 475$	
	Gayatri Tower	-			
	Adikara	2	1	HP Level 1 : 400 maka $2 \times 400 = 800$	
			2	HP Level 2 : 460 maka $2 \times 460 = 920$	
	Kamandaka	2	1	HP Level 1 : 450 maka $2 \times 450 = 900$	
			2	HP Level 2 : 510 maka $2 \times 510 = 1020$	
			3	HP Level 3 : 570 maka $2 \times 570 = 1140$	
			4	HP Level 4 : 630 maka $2 \times 630 = 1260$	
			5	HP Level 5 : 690 maka $2 \times 690 = 1380$	
			6	HP Level 6 : 750 maka $2 \times 750 = 1500$	
TH Level 3		Jumlah Bangunan	Level		
	Gandewa Tower	2	1	HP Level 1 : 325 maka $2 \times 325 = 650$	
			2	HP Level 2 : 375 maka $2 \times 375 = 750$	
			3	HP Level 3 : 425 maka $2 \times 425 = 850$	
			4	HP Level 4 : 475 maka $2 \times 475 = 950$	
			5	HP Level 5 : 525 maka $2 \times 525 = 1050$	
			6	HP Level 6 : 575 maka $2 \times 575 = 1150$	1150
	Gayatri Tower	1	1	HP Level 1 : 570 maka $1 \times 570 = 570$	
			2	HP Level 2 : 610 maka $1 \times 610 = 610$	
			3	HP Level 3 : 650 maka $1 \times 650 = 650$	
			4	HP Level 4 : 690 maka $1 \times 690 = 690$	690
	Adikara	2	1	HP Level 1 : 400 maka $2 \times 400 = 800$	
			2	HP Level 2 : 460 maka $2 \times 460 = 920$	
			3	HP Level 3 : 530 maka $2 \times 530 = 1060$	
			4	HP Level 4 : 580 maka $2 \times 580 = 1160$	1160
	Kamandaka	2	1	HP Level 1 : 450 maka $3 \times 450 = 1350$	
			2	HP Level 2 : 510 maka $3 \times 510 = 1530$	

			3	HP Level 3 : 570 maka $3 \times 570 = 1710$	
			4	HP Level 4 : 630 maka $3 \times 630 = 1890$	
			5	HP Level 5 : 690 maka $3 \times 690 = 2070$	
			6	HP Level 6 : 750 maka $3 \times 750 = 2250$	
			7	HP Level 7 : 810 maka $3 \times 810 = 2430$	
			8	HP Level 8 : 970 maka $3 \times 970 = 2910$	
			9	HP Level 9 : 1030 maka $3 \times 1030 = 3090$	3090
				JUMLAH MAKSIMAL	6090

Sedangkan untuk *damage* yang diterima musuh, memiliki perbedaan dengan *damage* atau kerusakan yang diterima oleh pahlawan maupun pasukan. Karena musuh yang merupakan *Tower Defense* ini tidak memiliki armor atau ketahanan terhadap serangan, maka *damage* yang akan diterimanya adalah murni *damage* yang diberikan oleh pahlawan dan pasukan. *Damage* yang diterima musuh untuk setiap pahlawan berbeda-beda, seperti contoh, jika pahlawannya Batara, maka minimal *damage* yang diberikan Batara adalah Batara memiliki *damage* bernilai 20, kemudian digabungkan atau ditambah dengan *damage* oleh pasukannya, bila minimal pasukan 1 maka *damage* yang dimiliki pasukan adalah 5 sehingga saat diakumulasi, *damage* yang akan diterima musuh adalah $20 + 5 = 25$. Berikut penjelasan rumus yang lebih detail beserta tabel setiap kondisi *damage received* musuh dari masing – masing pahlawan :

$$\text{Damage}_{\text{Min}} \text{ Musuh} = \text{Damage}_{\text{min}} \text{ Pahlawan} + (\text{jml pasukan} * \text{damage}_{\text{min}} \text{ pasukan}) \quad (3.6)$$

$$\text{Damage}_{\text{Max}} \text{ Musuh} = \text{Damage}_{\text{max}} \text{ Pahlawan} + (\text{jml pasukan} * \text{damage}_{\text{max}} \text{ pasukan}) \quad (3.7)$$

Tabel 3.9 Tabel *Damage Receive* Musuh

Pahlawan	Damage Lawan
Batara	1250
	25
	Damage Lawan
Wira	900
	12
	Damage Lawan
Kirna	66
	25

Semua parameter yang dibutuhkan untuk dapat mengoptimasi perilaku pahlawan telah dihitung dan telah tersajikan dalam tabel – tabel. Mulai dari Kesehatan minimum dan maksimum Pahlawan, Kesehatan minimum dan maksimum Musuh, Kesehatan minimum dan maksimum Pasukan, *Damage Received* Pahlawan, *Damage Received* Musuh, serta *Damage Received* Pasuka, semuanya berjumlah 6 parameter. Keenam parameter yang memiliki *range* minimum dan maksimum inilah nantinya yang akan diproses oleh NSGA-II, sebanyak populasi dan generasi yang diinginkan, semakin banyak populasi dan generasi, maka semakin bagus optimasinya.

3.7 Optimasi Multi Objective

Mengoptimasi perilaku pahlawan yang memiliki banyak tujuan atau multi objective menggunakan metode optimasi yaitu NSGA-II. NSGA-II digunakan untuk melakukan optimasi terhadap individu-individu untuk membangun solusi-solusi berdasarkan obyektif-obyektifnya. Pada metode ini, individu-individu yang dibangkitkan untuk level berikutnya disimulasikan dengan melawan setiap komposisi musuh yang ditemui. Hasil dari simulasi tersebut bertujuan untuk mendapatkan nilai obyektif dari setiap individu. NSGA-II bertujuan untuk mengoptimisasi individu-individu dan mencari sekumpulan solusi terbaik yang nantinya dipilih salah satunya untuk digunakan pada level selanjutnya [27]. Optimisasi yang dicari pada NSGA-II adalah optimisasi minimum dari obyektif-obyektif yang dimiliki dari setiap individu.

Metode NSGA-II ini sangat berbeda dengan Algoritma Genetika pada umumnya. Dimana solusi terbaik yang dihasilkan bukanlah solusi tunggal berdasarkan fitness terbaik, namun berupa sekumpulan solusi yang *tergenerate* pada *pareto front* pertama berdasarkan dominasi. Sebuah keputusan yang *non-dominan* adalah keputusan yang tidak di dominasi oleh keputusan lain. Bisa jadi dalam sebuah kasus, solusi yang terbaik terletak pak solusi yang *non-dominan* tersebut. Sehingga metode NSGA-II ini dinilai sebagai metode terbaik untuk menemukan sebuah solusi yang tidak di dominasi oleh solusi yang lain, dan keputusan dari luar dibutuhkan untuk memilih salah satu dari solusi-solusi terbaik yang dihasilkan. Hal tersebut dapat dilakukan karena sekumpulan solusi yang

dihasilkan adalah solusi-solusi yang sama baiknya, sehingga pemilihan solusi secara acak pun dimungkinkan. Bagian berikutnya akan menjelaskan setiap komponen yang dikembangkan dalam NSGA-II untuk penelitian ini berdasarkan permasalahan yang dihadapi. NSGA-II yang merupakan metode untuk *multi objective* pada 2-3 kasus pencarian fungsi objective, pada penelitian ini NSGA-II digunakan untuk mencari 5 fungsi objective. Perlu diingat bahwa desain NSGA-II merupakan problem dependant (bergantung kepada permasalahan) dan Sub sub bab berikut mempresentasikan NSGA-II yang digunakan untuk pendekatan yang diajukan [27]. Diagram alir NSGA – II pada Gambar 2.10 akan menjelaskan langkah – langkah NSGA-II untuk mengoptimasi kelima perilaku pahlawan.

Dengan pembangkitan populasi dan generasi dari beberapa parameter untuk mengevaluasi setiap nilai fungsi objective, maka akan dihasilkan *output* berupa lima perilaku pahlawan yang optimal, dimana saat pengujian atau simulasi akan digunakan pahlawan untuk menentukan perilaku manakah yang paling optimal ketika dihadapkan dengan berbagai macam musuh, mulai dari musuh yang rendah atau paling lemah, hingga musuh yang paling tinggi atau paling kuat. Langkah awal dalam memproses perilaku pahlawan yang multiobjective ini adalah menentukan berapa populasi dan generasi dimana merupakan iterasi sebanyak populasi yang diinputkan. Generasi tersebut dibangkitkan secara acak pada awal permainan.

3.7.1 Populasi Awal

Populasi awal dibangkitkan secara acak 100 populasi. Populasi dapat dikatakan sebagai skenario. Pembangkitan populasi awal sebesar 100 bertujuan memberikan banyak variasi atau pilihan skenario untuk memunculkan banyak variasi perilaku pahlawan.

3.7.2 Evaluasi Fungsi Objective

Untuk setiap perilaku atau fungsi objective, terdapat 5 fungsi objective yang masing – masing berbeda berdasarkan kondisi untuk setiap parameter *evaluate objective function*. Terdapat 6 parameter *evaluate objective function* yang *range*-nya (min-max) akan *dirandom* sebanyak N populasi hingga menghasilkan 1 perilaku pahlawan yang paling optimal. Berikut 6 parameter yang berpengaruh

besar terhadap penentuan perilaku yang optimal dari 5 tujuan atau multi objective pahlawan ini :

1. **Health Pahlawan** : Kesehatan pahlawan dengan *range* min dan max
2. **Health Musuh** : Kesehatan musuh dengan *range* min dan max
3. **Health Pasukan** : Kesehatan pasukan dengan *range* min dan max
4. **Damage Received Pahlawan** : *Damage* yang diterima pahlawan dengan *range* min dan max
5. **Damage Received Musuh** : *Damage* yang diterima pahlawan dengan *range* min dan max
6. **Damage Received Pasukan** : *Damage* yang diterima pahlawan dengan *range* min dan max

Dari keenam parameter tersebut maka diperoleh rumus *evaluate objective function* untuk masing – masing pahlawan, berikut rumus – rumusnya :

Inisialisasi untuk setiap parameter pada *evaluate objective function* :

<i>Health</i> Pahlawan	= x(1)	<i>Damage Received</i> Pahlawan	= x(4)
<i>Health</i> Musuh	= x(2)	<i>Damage Received</i> Musuh	= x(5)
<i>Health</i> Pasukan	= x(3)	<i>Damage Received</i> Pasukan	= x(6)

Pendekatan yang digunakan untuk membangun fungsi objective ini adalah operasi aritmatika biasa, perkalian, penambahan, pengurangan, dan pembagian. Pada pendekatan ini fungsi matematis untuk mendeskripsikan pareto-optimal front diasumsikan ke dalam ruang obyektif dan keseluruhan ruang pencarian obyektif dibangun berdasarkan front ini. Pada penelitian ini, kondisi Pareto-optimal front adalah dimana semua fungsi obyektif berbanding lurus dari tiap perilaku pahlawan (*sgresif*, *supportif*, *leadership*, *egois 1*, dan *egois 2*) dari setiap pahlawan dengan hasil simulasi tiap pahlwan terhadap berbagai macam musuh yang ditemui oleh pemain. Dari keenam parameter yang disebut dengan *decision variable* maka diperoleh rumus untuk setiap fungsi objective *agresif*, *supportif*, *leadership*, *egois1*, dan *egois 2* untuk masing – masing pahlawan.

Batara Maheswara :

Perilaku AGRESIF

$$f(1) = ((180*x(1))-(2.6*x(2))+(0.7*x(3))-(0.2*x(4))+(1.5*x(5))-(0.2*x(6)))/6; \quad (3.8)$$

Perilaku SUPPORTIF

$$f(2) = ((135*x(1))-(1.3*x(2))-(0.4*x(3))+(0.3*x(4))-(1*x(5))-(2.5*x(6)))/6; \quad (3.9)$$

Perilaku LEADERSHIP

$$f(3) = ((90*x(1))-(1.5*x(2))+(1*x(3))+(1.2*x(4))-(1.3*x(5))+(0.5*x(6)))/6; \quad (3.10)$$

Perilaku EGOIS 1

$$f(4) = ((90*x(1))-(2.5*x(2))-(0*x(3))+(20*x(4))-(2*x(5))-(0*x(6)))/6; \quad (3.11)$$

Perilaku EGOIS 2

$$f(5) = ((180*x(1))-(5*x(2))+(0.2*x(3))+(40*x(4))-(4*x(5))+(0.2*x(6)))/6; \quad (3.12)$$

Wira Oragastra :**Perilaku AGRESIF**

$$f(1) = ((180*x(1))-(2.6*x(2))+(0.7*x(3))-(0.2*x(4))+(1.5*x(5))-(0.2*x(6)))/6; \quad (3.13)$$

Perilaku SUPPORTIF

$$f(2) = ((180*x(1))-(1.5*x(2))-(0.5*x(3))+(0.6*x(4))-(1.2*x(5))-(1.5*x(6)))/6; \quad (3.14)$$

Perilaku LEADERSHIP

$$f(3) = ((90*x(1))-(2*x(2))+(1.2*x(3))+(1.4*x(4))-(1.1*x(5))+(0.2*x(6)))/6; \quad (3.15)$$

Perilaku EGOIS 1

$$f(4) = ((120*x(1))-(2.5*x(2))-(0*x(3))+(20*x(4))-(1.5*x(5))-(0*x(6)))/6; \quad (3.16)$$

Perilaku EGOIS 2

$$f(5) = ((120*x(1))-(3*x(2))+(0.2*x(3))+(40*x(4))-(4*x(5))+(6*x(6)))/6; \quad (3.17)$$

Kirna Waranggani :**Perilaku AGRESIF**

$$f(1) = ((120*x(1))-(2.5*x(2))+(1.2*x(3))-(0.1*x(4))+(6*x(5))-(0.1*x(6)))/6; \quad (3.18)$$

Perilaku SUPPORTIF

$$f(2) = ((180*x(1))-(1.5*x(2))-(0.2*x(3))+(0.5*x(4))-(1.2*x(5))-(1.6*x(6)))/6; \quad (3.19)$$

Perilaku LEADERSHIP

$$f(3) = ((225*x(1))-(4*x(2))+(3*x(3))+(5*x(4))-(2.4*x(5))+(0.6*x(6)))/6; \quad (3.20)$$

Perilaku EGOIS 1

$$f(4) = ((90*x(1))-(1.5*x(2))-(0*x(3))+(20*x(4))-(1.5*x(5))-(0*x(6)))/6; \quad (3.21)$$

Perilaku *EGOIS 2*

$$f(5) = ((120 \cdot x(1)) - (1.5 \cdot x(2)) + (0.4 \cdot x(3)) + (20 \cdot x(4)) - (9 \cdot x(5)) + (2.4 \cdot x(6))) / 6; \quad (3.22)$$

Sebagai contoh, adalah rumus fungsi objective Batara untuk perilaku *agresif* yaitu $f(1) = ((180 \cdot x(1)) - (2.6 \cdot x(2)) + (0.7 \cdot x(3)) - (0.2 \cdot x(4)) + (1.5 \cdot x(5)) - (0.2 \cdot x(6))) / 6$; dimana $f(1)$ merupakan fungsi objective *agresif* yang didapat dari hasil operasi aritmatika perkalian antara 180 dengan $x(1)$ yaitu parameter *health* pasukan. NSGA-II akan membangkitkan acak setiap *range* parameter yang telah diinputkan. *Range* untuk parameter $x(1)$ atau *health* pahlawan adalah 100 – 150. Karena perilaku *agresif* dapat didapat bila kondisi kesehatan pahlawan harus lebih besar daripada musuhnya, maka parameter *health* pahlawan ini harus dikalikan dengan bilangan yang besar. Begitupun dengan *health* musuh, maka *health* musuh harus dikalikan dengan bilangan yang lebih kecil yaitu 2,6.

Semakin besar pengaruhnya sebuah parameter terhadap perilaku yang dipilih, maka bilangan yang dikalikan untuk parameter tersebut harus diperbesar ataupun diperkecil nilainya. Operasi aritmatika diperhatikan dengan melihat kondisi parameter apa yang dibandingkan setelahnya. Misal : untuk menampilkan perilaku yang *agresif*, parameter yang pertama dilihat adalah *health* pahlawan, dalam penelitian ini *health* pahlawan hanya bernilai 100-150 sedangkan parameter kedua yang kita lihat setelahnya yaitu musuh, musuh memiliki nilai kesehatan yang jauh lebih besar daripada pahlawan, nilainya berkisar antara 325-6090. Maka dengan membandingkan kedua parameter tersebut, jelas yang harus kita lakukan adalah memperbesar perkalian parameter kesehatan pahlawan, dan mengecilkan parameter kesehatan musuh. Begitupun langkah selanjutnya, melihat kondisi kesehatan pasukan, kemudian *damage* yang diterima pahlawan itu sendiri, *damage* yang diterima musuhnya, serta *damage* yang diterima oleh pasukannya.

3.7.3 Pengurutan Non-Dominasi (*Non-dominated Sort*)

Suatu individu dapat dikatakan mendominasi individu lainnya jika memenuhi aturan-aturan berikut:

1. Individu A tidak lebih buruk ketimbang individu B pada semua obyektif ($A \geq B$).
2. Individu A setidaknya memiliki satu obyektif lebih baik ketimbang individu B.

Berdasarkan aturan tersebut, setiap individu dibandingkan dengan individu lainnya di dalam satu populasi. Permasalahan optimisasi dalam kasus ini adalah mencari nilai minimum dari 5 obyektif sehingga kondisi dominasi adalah jika suatu individu tidak lebih buruk dari individu lainnya dan setidaknya memiliki satu obyektif yang nilainya lebih besar daripada individu lainnya (misal, Obyektif A.1 \geq Obyektif B.1,...,Z.1 AND Obyektif A.2 \geq Obyektif B.2,...,Z.2 AND Obyektif A.3 $>$ Obyektif B.3,...,Z.3). Individu tersebut akan berada pada front pertama. Kemudian untuk mengisi front berikutnya berdasarkan individu yang didominasi oleh front sebelumnya.

3.7.4 Crowding Distance

Setelah pengurutan berdasarkan non-dominasi dilakukan, maka perhitungan selanjutnya adalah *crowding distance*. Dimana individu dipilih berdasarkan peringkat dan *crowding distance*. Sehingga semua individu pada populasi akan diberikan nilai *crowding distance*. *Crowding distance* tiap individu dihitung berdasarkan *front* individu tersebut berada, hanya individu – individu yang berada di *front* yang akan dihitung *crowding distance* nya. Sehingga membandingkan *crowding distance* antara dua individu yang berbeda *front* adalah hal yang sia - sia.

Ide dasar dibalik *crowding distance* adalah menemukan *euclidian distance* antara setiap individu pada *front* berdasarkan jumlah obyektifnya pada ruang dimensi sesuai dengan jumlah obyektif. Persamaan untuk mendapatkan *crowding distance* adalah sebagai berikut:

$$I(d_k) = I(d_k) + \frac{I(k+1).m - I(k-1).m}{f_m^{max} - f_m^{min}} \quad (3.23)$$

Dimana $I(d_k)$ adalah *crowding distance* dari individu ke-k, k adalah indeks individu pada *front* F_i yang dimulai dari 2 hingga n-1. Untuk $I(d_1)$ dan $I(d_n)$ yang menjadi nilai batas untuk individu pada *front* F_i diberikan nilai tak hingga (∞). $I(k).m$ adalah nilai dari fungsi obyektif ke-m pada individu ke-k, sedangkan f_m^{max} dan f_m^{min} secara berturut adalah fungsi obyektif terbesar dan terkecil dari fungsi

obyektif ke- m pada front yang dilakukan *crowding distance*. Untuk $I(d_1)$ dan $I(d_n)$ yang menjadi nilai batas untuk individu pada front F_i diberikan nilai tak hingga (∞).

3.7.5 Offspring (Keturunan)

Langkah selanjutnya adalah mencari keturunan yang baik dimana untuk mencari keturunan yang baik terdapat beberapa langkah yang didasarkan pada GA yaitu seleksi, persilangan, dan mutasi.

3.7.5.1 Seleksi

Saat individu telah diurutkan berdasarkan non-dominasi dan *crowding distance* telah dihitung, seleksi dilakukan dengan *crowded-comparison-operator* (\prec_n). Perbandingan (*comparison*) dilakukan dengan aturan sebagai berikut:

1. Peringkat non-dominasi (prank). Sebagai contoh, jika individu pada front F_i akan memiliki peringkat $\text{prank} = i$.
2. Crowding Distance $F_i(d_j)$
 $p \prec_n q$ jika:
 - $\text{prank} < \text{qrank}$
 - atau jika p dan q berada pada front yang sama (F_i) maka dicari perbandingan *crowding distance* $F_i(d_p) > F_i(d_q)$.

3.7.5.2 Crossover (Persilangan)

Pada penelitian ini, nilai yang digunakan pada NSGA-II bertipe Real sehingga persilangan yang digunakan adalah *Simulated Binary Crossover* (SBX) [38].

Simulated Binary Crossover

Setiap individu baru pada populasi berikut berasal dari sepasang individu dari generasi saat ini yang mengalami persilangan (*crossover*). Proses ini akan menghasilkan dua offspring atau individu “anak”. Ide dasar dari proses persilangan ini adalah offspring akan membawa karakteristik dari dua individu “orang tua” ke generasi berikutnya. Metode persilangan yang digunakan pada penelitian ini adalah *Simulated Binary Crossover*.

Proses yang dilakukan SBX adalah menyilangkan bilangan real dengan mensimulasikan seperti persilangan biner yang diobservasi secara alami dan ditunjukkan sebagai berikut.

$$\begin{aligned} o_{1,k} &= \frac{1}{2}[(1 - \beta_k)p_{1,k} + (1 + \beta_k)p_{2,k}] \\ o_{2,k} &= \frac{1}{2}[(1 + \beta_k)p_{1,k} + (1 - \beta_k)p_{2,k}] \end{aligned} \quad (3.24)$$

dimana $o_{i,k}$ adalah offspring i^{th} dengan komponen k^{th} . $p_{i,k}$ adalah parent terpilih dan $\beta_k (\geq 0)$ adalah sampel dari angka yang dibangkitkan secara acak dengan kepadatan

$$\begin{aligned} p(\beta) &= \frac{1}{2}(\eta_c + 1)\beta^{\eta_c}, \text{ if } 0 \leq \beta \leq 1 \\ p(\beta) &= \frac{1}{2}(\eta_c + 1)\frac{1}{\beta^{\eta_c+2}}, \text{ if } \beta > 1 \end{aligned} \quad (3.25)$$

Distribusi ini bisa didapatkan dari angka acak yang disampel secara *uniform* u berkisar antara (0,1). η_c adalah indeks distribusi dari persilangan. Indeks distribusi ini bertujuan untuk mengukur seberapa baik penyebaran dari *offspring* dari *parent*-nya. η_c yang digunakan pada penelitian ini sebesar 20, 50, 75, dan 100.

3.7.5.3 Mutasi

Pada penelitian ini, nilai yang digunakan pada NSGA-II bertipe Real sehingga mutasi yang digunakan *polynomial mutation* [39].

Polynomial Mutation

Mutasi adalah teknik yang umum digunakan pada algoritma genetika untuk menjaga keragaman genetik di dalam populasi saat melalui generasi. Tingkat mutasi yang terlalu rendah akan mengarahkan pada kondisi di mana setiap individu akan mirip dan keragaman solusi akan hilang. Terlalu besarnya tingkat mutasi maka dapat menyebabkan hilangnya solusi yang baik.

Penelitian ini menggunakan suatu model mutasi yang menerapkan model probabilitas kepada solusi. Model mutasi ini disebut dengan Polynomial Mutation.

$$o_k = p_k + (p_k^u - p_k^l)\delta_k \quad (3.26)$$

dimana o_k adalah offspring dan p_k adalah parent dengan p_k^u menjadi batas atas pada komponen parent. p_k^l adalah batas bawah dan δ_k adalah variasi kecil yang mana dikalkulasi dari distribusi polinomial menggunakan :

$$\begin{aligned} \delta_k &= (2r_k)^{\frac{1}{\eta_m+1}} - 1, \text{ if } r_k < 0.5 \\ \delta_k &= 1 - [2(1 - r_k)]^{\frac{1}{\eta_m+1}}, \text{ if } r_k \geq 0.5 \end{aligned} \quad (3.27)$$

r_k adalah angka acak yang disampel secara uniform berkisar antara (0,1) dan η_m adalah indeks distribusi mutasi. η_m yang digunakan pada penelitian ini adalah 20, 50, 75, dan 100 serta probabilitas mutasi adalah $1/n$ dimana n adalah ukuran populasi.

Pada bab selanjutnya yaitu hasil dan pembahasan akan ditunjukkan pareto optimal setelah proses seleksi ini dilakukan, dimana pareto optimal tersebut mewakili dua tujuan yang saling bertentangan yaitu kondisi dimana pahlawan mati namun pasukan tetap hidup (*supportif*) dan pahlawan yang tetap hidup namun pasukannya mati (*agresif*). Dilanjutkan dengan pengujian dari kelima objective yang berbeda untuk mendapat perilaku optimal dari setiap pahlawan.

3.8 Skenario Pengujian

Sebuah scenario pengujian dilakukan guna mengetahui apakah proses dari metode yang diterapkan telah sesuai dengan hasil yang diharapkan. Dalam skenario pengujian ini, akan diuji masing – masing pahlawan saat bertemu berbagai macam kondisi musuh.

1. Pengujian Performa NSGA – II untuk dua objective yang berbeda setiap pahlawan, yaitu antara kondisi pahlawan yang mati namun pasukannya tetap hidup (*supportif*) dan kondisi pahlawan yang tetap hidup untuk memenangkan pertempuran namun pasukannya mati (*agresif*).
 - a. Pengujian 100 Populasi dan 20 Generasi
 - b. Pengujian 100 Populasi dan 50 Generasi
 - c. Pengujian 100 Populasi dan 75 Generasi
 - d. Pengujian 100 Populasi dan 100 Generasi

Pengujian Performa NSGA – II untuk perilaku optimal setiap pahlawan dengan lima objective yang berbeda.

BAB 4

HASIL DAN PEMBAHASAN

Bab ini memberikan penjelasan mendalam mengenai pengujian yang dilakukan untuk mengevaluasi pendekatan yang dikembangkan dan memberikan hasil yang didapatkan disertai dengan analisa. Dua model pengujian dilakukan pada bab ini, yaitu:

Pengujian 1 : Simulasi untuk melihat performa NSGA-II untuk dua objective yang berbeda.

Pengujian 2 : Simulasi untuk melihat performa NSGA-II untuk lima objective yang berbeda.

Pengujian 3 : Simulasi untuk membandingkan keberhasilan dalam penyerangan ketika terdapat pemilihan perilaku yang optimal dan tidak.

Pengujian 1 melibatkan beberapa kali percobaan untuk mendapatkan konfigurasi NSGA-II yang ideal melalui simulasi dan melihat performa dari NSGA-II. Hasil yang dianggap paling baik dan efisien akan digunakan di dalam permainan. Pengujian 2 hanya dilakukan 1 kali percobaan yaitu untuk jumlah populasi dan generasi yang paling stabil. Seperti yang telah dibuat pada Bab 3, beberapa skenario pengujian ini untuk menguji kemampuan parameter yang telah ditentukan dari pahlawan terhadap beberapa musuh yang ditemuinya.

4.1 Pengujian Performa NSGA - II

Simulasi dilakukan dengan membangun beberapa model jumlah populasi yaitu 100 individu dengan berbagai macam generasi, 20, 50, 75, dan 100 dari setiap pahlawan yaitu Batara, Wira, dan Kirna. Dalam permainan DWIPA YUDHA yang menjadi pengujian terdapat 6 variabel yang menjadi variabel keputusan dalam penelitian ini adalah *health* pahlawan, *health musuh*, *health pasukan*, *damage received* pahlawan, *damage received* pasukan, serta *damage received* musuh. Setiap rumus fungsi objective telah dijelaskan sebelumnya pada Sub Bab 3.7.2.

4.1.1 Pengujian Dua Fungsi Objective Batara dengan 100 Populasi dan 20 Generasi

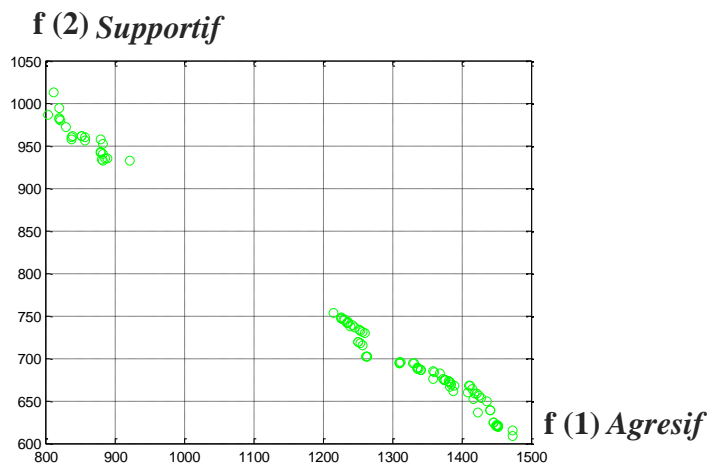
Pada pengujian pertama, untuk Pahlawan Batara Maheswara dengan dua fungsi objective yang berbeda. Berikut Tabel Pengujiannya dengan dua objective yang berbeda (*agresif* dan *supportif*) menggunakan NSGA-II :

Tabel 4.1 Tabel Dua Fungsi Objective Batara 100 Populasi 20 Generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)
96	105.4932	5734.274	2580.261	60.81125	1041.426	110.4674	1235.621	742.5943
97	105.5582	5735.878	2691.074	58.64595	1072.114	112.1975	1257.492	730.3792
98	105.4366	5727.155	2502.121	53.93346	1055.651	111.1173	1231.655	745.0874
99	105.4133	5695.134	3321.63	52.29082	1225.155	102.169	1383.171	672.2629
100	100.0002	5363.769	1187.579	134.5678	179.5871	57.69037	852.7443	961.4415
Jumlah							76	24

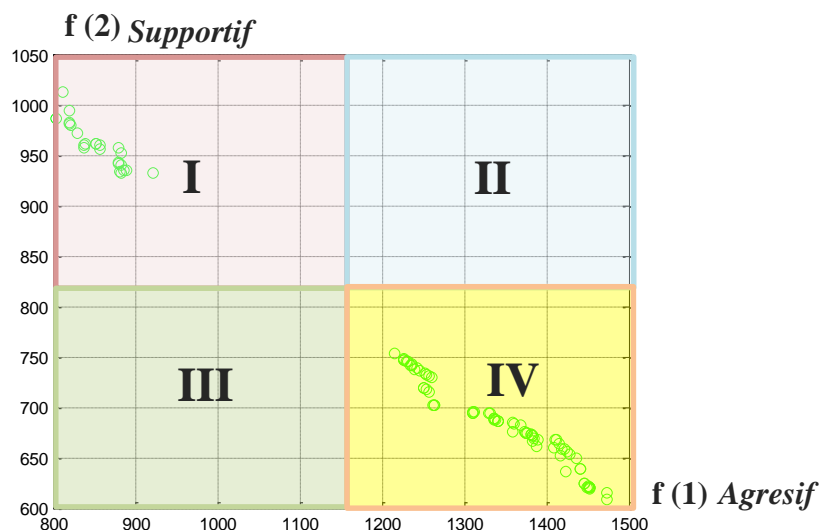
Tabel diatas akan ditunjukkan secara lengkap pada Lampiran 1. Dari beberapa yang diambil dari Tabel 4.1 pada Lampiran 1 didapatkan bahwa seluruh datanya menunjukkan antara dua objective yang berbeda yang merupakan tujuan dari penelitian ini, yaitu antara pahlawan yang tidak hanya dapat menyerang menghancurkan musuhnya (*agresif*) namun juga dapat melindungi pasukannya sehingga pasukannya tersebut masih tetap hidup (*supportif*) meskipun ia sendiri harus mati atau mengorbankan nyawanya. Dari keseluruhan data, 100 populasi dengan 20 generasi, didapatkan seluruh populasi menunjukkan Batara yang sangat *agresif*, karena didapatkan f(1) atau perilaku *agresif* lah yang paling banyak muncul (berjumlah 76 dari 100 populasi).

Seluruh data tersebut akan membentuk sebuah *pareto front* antara dua objective yang berbeda atau saling bertentangan tersebut. Dimana grafik *pareto front* disebut pula *pareto optimal* yaitu kumpulan solusi yang paling optimal dari setiap kondisi yang ditemui oleh Batara. Grafik *pareto optimal* ditunjukkan oleh Gambar 4.1.



Gambar 4.1 *Pareto Optimal* Batara dengan 100 Populasi 20 Generasi

Dari grafik diatas didapat bahwa terjadi penyebaran titik pada *pareto front*, dimana titik – titik tersebut adalah sekumpulan solusi dari setiap kondisi yang ada pada 100 populasi. Populasi dapat dikatakan sebagai individu atau dalam *game* dapat kita sebut sebagai skenario. Jadi dari 100 populasi atau 100 skenario pengujian dengan 20 generasi, 20 generasi adalah 20 kali pembangkitan ulang generasi yang lebih baik, sehingga didapatkan kesimpulan dari grafik tersebut bahwa solusi optimal tersebar secara terpisah. Penyebaran yang terpisah tersebut agar lebih mudah membacanya, maka akan dibagi menjadi 4 kuadran. berikut tampilan grafik yang telah terbagi menjadi 4 kuadran, ditunjukkan oleh Gambar 4.2.



Gambar 4.2 Pembagian 4 kuadran *pareto front* 100 populasi 20 generasi

Maka didapatkan grafik seperti Gambar 4.2 diatas. Grafik tersebut dibagi menjadi 4 kuadran. Kuadran I berwarna merah dimana titik – titik solusi optimal tersebar, menunjukkan perilaku *supportif* yang sangat tinggi namun perilaku *agresif* yang sangat rendah, sedangkan kuadran II yang berwarna biru tidak terdapat satupun titik solusi optimal yang tersebar, menunjukkan tidak ada perilaku *supportif* yang tinggi dan perilaku *agresif* yang tinggi pula untuk pahlawan. Untuk kuadran III yang berwarna hijau yang tidak ada pula titik solusi optimal yang tersebar, menunjukkan perilaku *supportif* yang rendah dan perilaku *agresif* yang juga rendah bukan perilaku optimal untuk pahlawan. Sedangkan pada kuadran IV yang berwarna kuning inilah titik – titik solusi optimal tersebar kembali yaitu pada perilaku *supportif* yang sangat rendah namun perilaku *agresif* yang sangat tinggi. Dari *pareto optimal* dengan 100 populasi 20 generasi, didapatkan titik – titik solusi optimal paling banyak tersebar pada kuadran IV yang menunjukkan bahwa perilaku optimal untuk pahlawan Batara adalah perilaku yang sangat tinggi perilaku *Agresif* nya dan sangat rendah perilaku *supportif* nya.

4.1.2 Pengujian Dua Fungsi Objective Batara dengan 100 Populasi dan 50 Generasi

Pada pengujian kedua, untuk Pahlawan Batara Maheswara dengan dua fungsi objective yang berbeda. Berikut Tabel Pengujiannya dengan dua objective yang berbeda (*agresif* dan *supportif*) menggunakan NSGA-II :

Tabel 4.2 Tabel Dua Fungsi Objective Batara 100 Populasi 50 Generasi

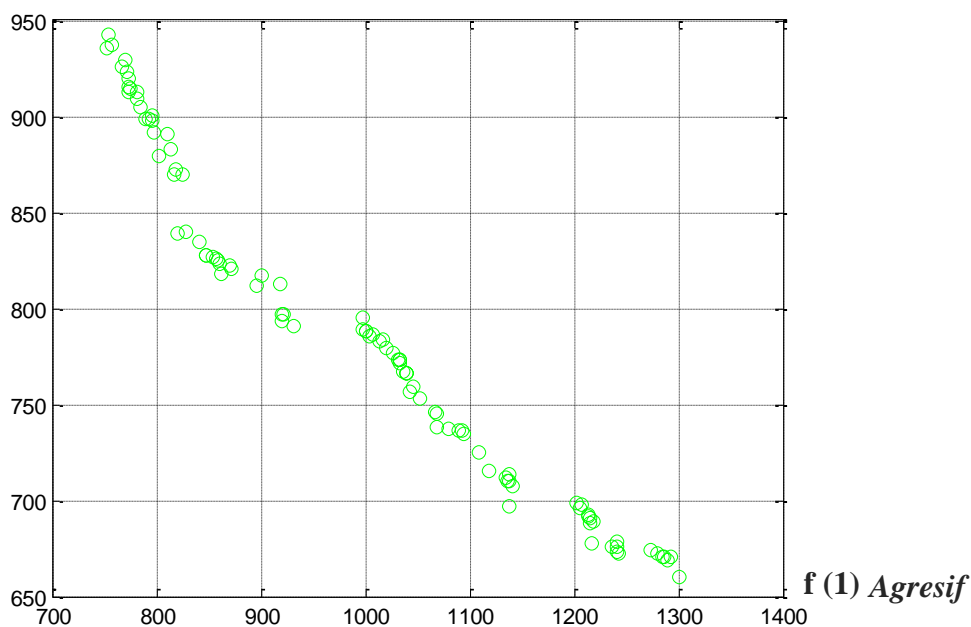
Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)
96	102.6577	5563.945	3809.443	135.5572	423.0983	213.4187	1207.264	697.6501
97	104.5129	5807.773	1588.652	128.1444	506.5432	232.3699	918.648	812.4427
98	104.1955	5820.826	1296.752	128.3138	506.324	230.8346	869.4056	822.6179
99	104.3271	5819.997	1683.524	127.4592	514.9218	234.5864	920.8888	796.9346
100	104.0412	5738.632	582.3007	122.548	477.2456	226.6728	810.1018	890.8766
Jumlah							74	26

Tabel diatas akan ditunjukkan secara lengkap pada Lampiran 2. Dari beberapa yang diambil dari Tabel 4.2 pada Lampiran 2 didapatkan bahwa seluruh

datanya menunjukkan antara dua objective yang berbeda yang merupakan tujuan dari penelitian ini, yaitu antara pahlawan yang tidak hanya dapat menyerang menghancurkan musuhnya (*agresif*) namun juga dapat melindungi pasukannya sehingga pasukannya tersebut masih tetap hidup (*supportif*) meskipun ia sendiri harus mati atau mengorbankan nyawanya. Dari keseluruhan data, 100 populasi dengan 50 generasi, didapatkan seluruh populasi menunjukkan Batara yang sangat *agresif*, karena didapatkan $f(1)$ atau perilaku *agresif* lah yang paling banyak muncul (berjumlah 74 dari 100 populasi).

Seluruh data tersebut akan membentuk sebuah *pareto front* antara dua objective yang berbeda atau saling bertentangan tersebut. Dimana grafik *pareto front* disebut pula *pareto optimal* yaitu kumpulan solusi yang paling optimal dari setiap kondisi yang ditemui oleh Batara. Grafik *pareto optimal* ditunjukkan oleh Gambar 4.3.

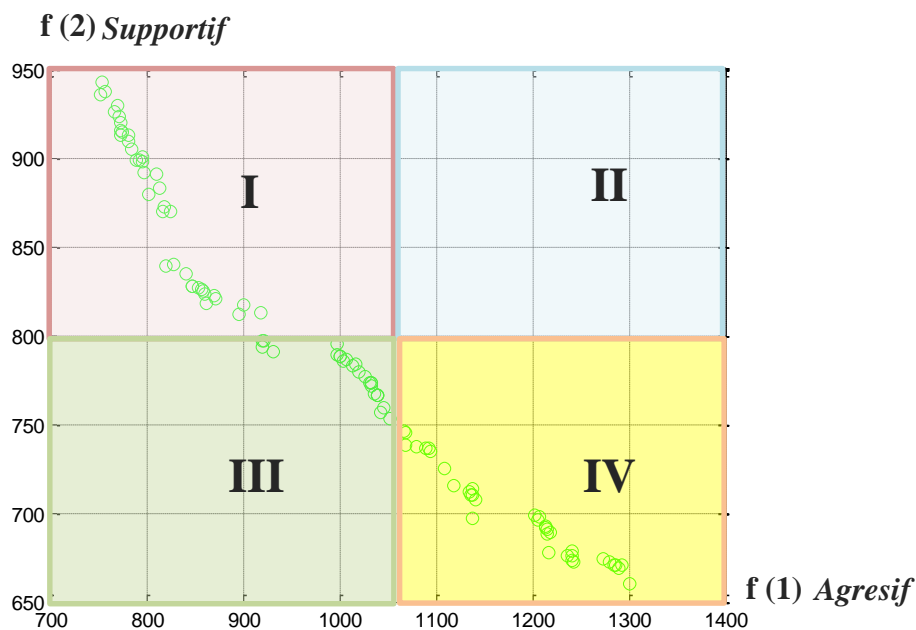
f (2) Supportif



Gambar 4.3 *Pareto Optimal* Batara dengan 100 Populasi 50 Generasi

Dari grafik diatas didapat bahwa terjadi penyebaran titik pada *pareto front*, dimana titik – titik tersebut adalah sekumpulan solusi dari setiap kondisi yang ada pada 100 populasi. Populasi dapat dikatakan sebagai individu atau dalam *game* dapat kita sebut sebagai skenario. Jadi dari 100 populasi atau 100 skenario

pengujian dengan 50 generasi, 50 generasi adalah 50 kali pembangkitan ulang generasi yang lebih baik, sehingga didapatkan kesimpulan dari grafik tersebut bahwa dengan 50 kali pembangkitan ulang generasi yang lebih baik, terjadi penyebaran titik yang lebih baik. Karena pada generasi 20 titik tersebar secara terpisah jauh dan terklasifikasi sendiri yaitu hanya pada kuadran I dan kuadran IV, maka pada generasi 50 solusi optimal tersebar pada kuadran I, kuadran III, dan kuadran IV hampir membentuk *pareto front* atau *pareto optimal*. Penyebaran tersebut agar lebih mudah membacanya, maka akan dibagi menjadi 4 kuadran pula. berikut tampilan grafik yang telah terbagi menjadi 4 kuadran, ditunjukkan oleh Gambar 4.4.



Gambar 4.4 Pembagian 4 kuadran *pareto front* 100 populasi 50 generasi

Maka didapatkan grafik seperti Gambar 4.4 diatas. Grafik tersebut dibagi menjadi 4 kuadran. Kuadran I berwarna merah dimana titik – titik solusi optimal tersebar, menunjukkan perilaku *supportif* yang sangat tinggi namun perilaku *agresif* yang sangat rendah, disini titik – titik solusi optimal tersebut menyebar menuju kuadran III yang berwarna hijau. Kuadran III tersebar titik solusi optimal didalamnya, menunjukkan perilaku *supportif* yang sedikit rendah begitupun perilaku *agresif* nya yang juga sedikit rendah. Kuadran II yang berwarna biru tidak terdapat satupun titik solusi optimal yang tersebar, menunjukkan tidak adanya perilaku

supportif yang tinggi dan perilaku *agresif* yang tinggi pula untuk pahlawan. Sedangkan pada kuadran IV yang berwarna kuning inilah titik – titik solusi optimal tersebar kembali yaitu pada perilaku *supportif* yang sangat rendah namun perilaku *agresif* yang sangat tinggi. Dari *pareto optimal* dengan 100 populasi 50 generasi, didapatkan titik – titik solusi optimal paling banyak tersebar pada kuadran I yang menunjukkan bahwa perilaku optimal untuk pahlawan Batara adalah perilaku yang sangat tinggi *supportif* nya dan sangat rendah *agresif* nya juga memiliki perilaku pada kuadran III yang sedikit rendah perilaku *supportif* dan *agresif*nya. Serta memiliki pula perilaku pada kuadran IV yaitu sangat tinggi *agresif* nya namun sangat rendah *supportif* nya. Titik – titik solusi optimal perilaku pahlawan di ketiga kuadran telah membentuk *pareto optimal* untuk fungsi minimum NSGA-II.

4.1.3 Pengujian Dua Fungsi Objective Batara dengan 100 Populasi dan 75 Generasi

Pada pengujian ketiga, untuk Pahlawan Batara Maheswara dengan dua fungsi objective yang berbeda. Berikut Tabel Pengujiannya dengan dua objective yang berbeda (*agresif* dan *supportif*) menggunakan NSGA-II :

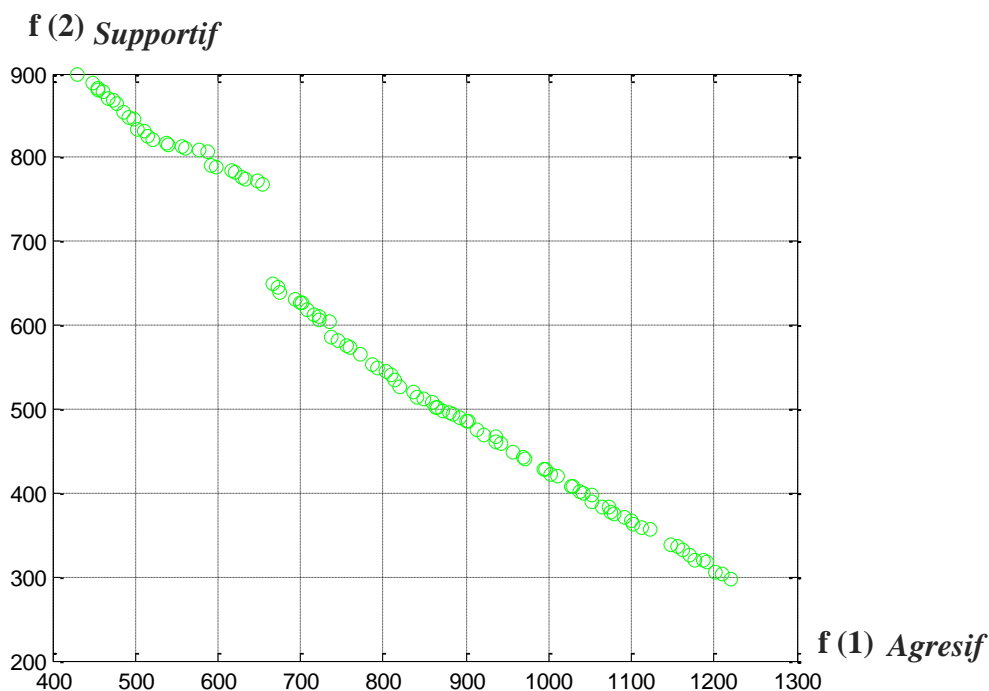
Tabel 4.3 Tabel Dua Fungsi Objective Batara 100 Populasi 75 Generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)
96	100	6037.059	1324.98	161.9563	794.2694	288	722.0907	609.358
97	100.0381	6083.182	2198.682	156.9517	1027.24	288	863.5899	502.8976
98	100.0054	6082.417	4479.912	142.7123	996.7359	285.141	1122.026	355.8072
99	102.4882	6089.528	40	276.6087	126.6233	229.0703	455.3184	881.2017
100	100	6081.315	3915.973	157.3623	1061.814	287.1208	1072.264	382.5823
Jumlah							71	29

Tabel diatas akan ditunjukkan secara lengkap pada Lampiran 3. Dari beberapa yang diambil dari Tabel 4.3 pada Lampiran 3 didapatkan bahwa seluruh datanya menunjukkan antara dua objective yang berbeda yang merupakan tujuan dari penelitian ini, yaitu antara pahlawan yang tidak hanya dapat menyerang menghancurkan musuhnya namun juga dapat melindungi pasukannya sehingga pasukannya tersebut masih tetap hidup (*supportif*) meskipun ia sendiri harus mati

atau mengorbankan nyawanya. Dari keseluruhan data, 100 populasi dengan 100 generasi, didapatkan seluruh populasi menunjukkan Batara yang sangat *agresif*, karena didapatkan $f(1)$ atau perilaku *agresif* lah yang paling banyak muncul (berjumlah 70 dari 100 populasi).

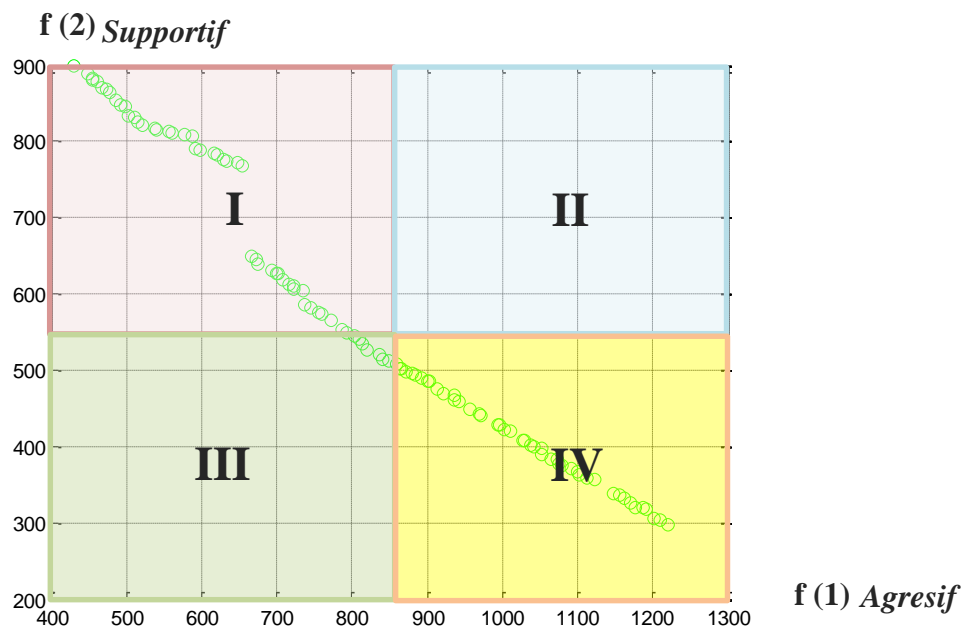
Seluruh data tersebut akan membentuk sebuah *pareto front* antara dua objective yang berbeda atau saling bertentangan tersebut. Dimana grafik *pareto front* disebut pula *pareto optimal* yaitu kumpulan solusi yang paling optimal dari setiap kondisi yang ditemui oleh Batara. Grafik *pareto optimal* ditunjukkan oleh Gambar 4.5.



Gambar 4.5 *Pareto Optimal* Batara dengan 100 Populasi 75 Generasi

Dari grafik diatas didapat bahwa terjadi penyebaran titik pada *pareto front*, dimana titik – titik tersebut adalah sekumpulan solusi dari setiap kondisi yang ada pada 100 populasi. Populasi dapat dikatakan sebagai individu atau dalam *game* dapat kita sebut sebagai skenario. Jadi dari 100 populasi atau 100 skenario pengujian dengan 75 generasi, 75 generasi adalah 75 kali pembangkitan ulang generasi yang lebih baik, sehingga didapatkan kesimpulan dari grafik tersebut bahwa dengan 75 kali pembangkitan ulang generasi yang lebih baik, terjadi penyebaran titik yang juga lebih baik. Bila pada generasi 50 titik tersebar titik solusi

optimal hampir membentuk *pareto front* atau *pareto optimal* untuk fungsi minimum NSGA-II maka pada generasi 75 titik solusi optimal juga telah membentuk *pareto optimal* dimana penyebaran titiknya lebih mendekati garis lurus, yang tersebar pada kuadran I, III dan kuadran IV. Penyebaran tersebut agar lebih mudah membacanya, maka akan dibagi menjadi 4 kuadran pula. berikut tampilan grafik yang telah terbagi menjadi 4 kuadran, ditunjukkan oleh Gambar 4.6.



Gambar 4.6 Pembagian 4 kuadran *pareto front* 100 populasi 75 generasi

Maka didapatkan grafik seperti Gambar 4.6 diatas. Grafik tersebut dibagi menjadi 4 kuadran. Kuadran I berwarna merah dimana titik – titik solusi optimal tersebar, menunjukkan perilaku *supportif* yang sangat tinggi namun perilaku *agresif* yang sangat rendah, disini titik – titik solusi optimal tersebut menyebar lebih luas hampir mendekati kuadran III yaitu perilaku *supportif* yang sedikit tinggi dan *agresif* yang sedikit tinggi pula. Kuadran II yang berwarna biru tidak terdapat satupun titik solusi optimal yang tersebar, menunjukkan tidak adanya perilaku *supportif* yang tinggi dan perilaku *agresif* yang tinggi pula untuk pahlawan. Untuk kuadran III yang berwarna hijau terdapat beberapa titik solusi optimal didalamnya, menunjukkan perilaku *supportif* yang sedikit tinggi dengan perilaku *agresif* nya yang mendekati nilai tinggi pula. Sedangkan pada kuadran IV yang berwarna kuning inilah titik – titik solusi optimal tersebar lagi yaitu pada perilaku *supportif*

yang sangat rendah namun perilaku *agresif* yang sangat tinggi. Dari *pareto optimal* dengan 100 populasi 75 generasi, didapatkan titik – titik solusi optimal hampir rata tersebar pada kuadran I, III, dan IV yang menunjukkan bahwa titik – titik solusi optimal hampir membentuk *pareto front* yang sempurna. Sehingga solusi perilaku optimal untuk pahlawan Batara tersebar diantara perilaku yang sangat tinggi *supportif* nya dan sangat rendah *agresif* nya, yang sedikit tinggi *supportif* dan *agresif* nya, serta yang sangat tinggi *agresif* nya dan sangat rendah *supportif* nya.

4.1.4 Pengujian Dua Fungsi Objective Batara dengan 100 Populasi dan 100 Generasi

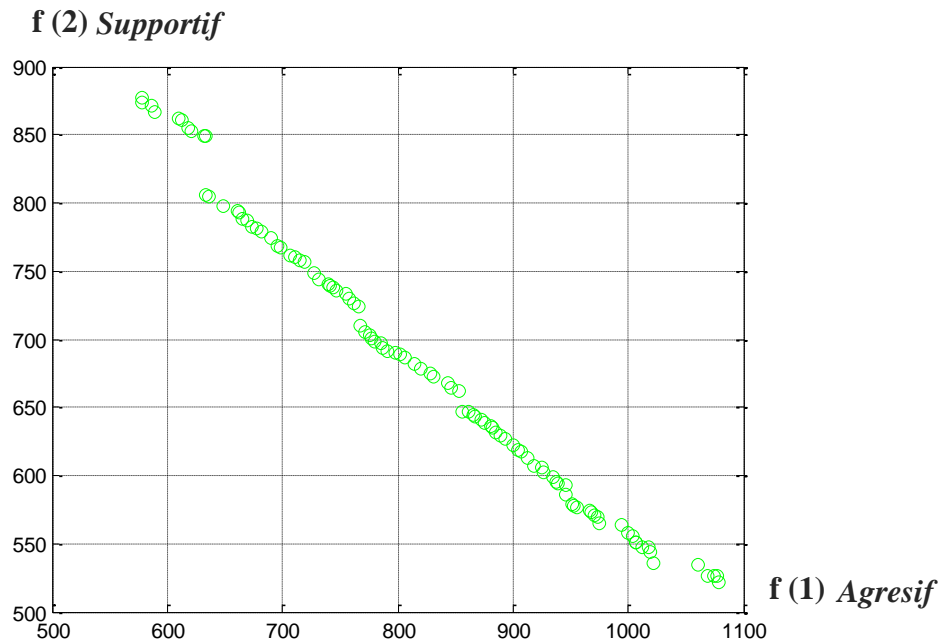
Pada pengujian keempat, untuk Pahlawan Batara Maheswara dengan dua fungsi objective yang berbeda. Berikut Tabel Pengujiannya dengan dua objective yang berbeda (*supportif* dan egois 1) menggunakan NSGA-II :

Tabel 4.4 Tabel Dua Fungsi Objective Batara 100 Populasi 100 Generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)
96	102.112	6083.876	3979.222	167.198	356.0171	215.1419	967.5148	573.4457
97	102.6879	6089.985	2121.621	168.3789	382.6984	212.9775	772.1277	705.4341
98	102.04	6085.445	3890.778	168.0977	352.1366	216.3626	953.3167	577.566
99	103.1221	6089.629	1741.035	171.1752	382.463	215.7226	740.6635	739.689
100	102.364	6087.586	3221.456	165.3995	371.9671	208.7692	889.3218	628.7373
Jumlah							70	30

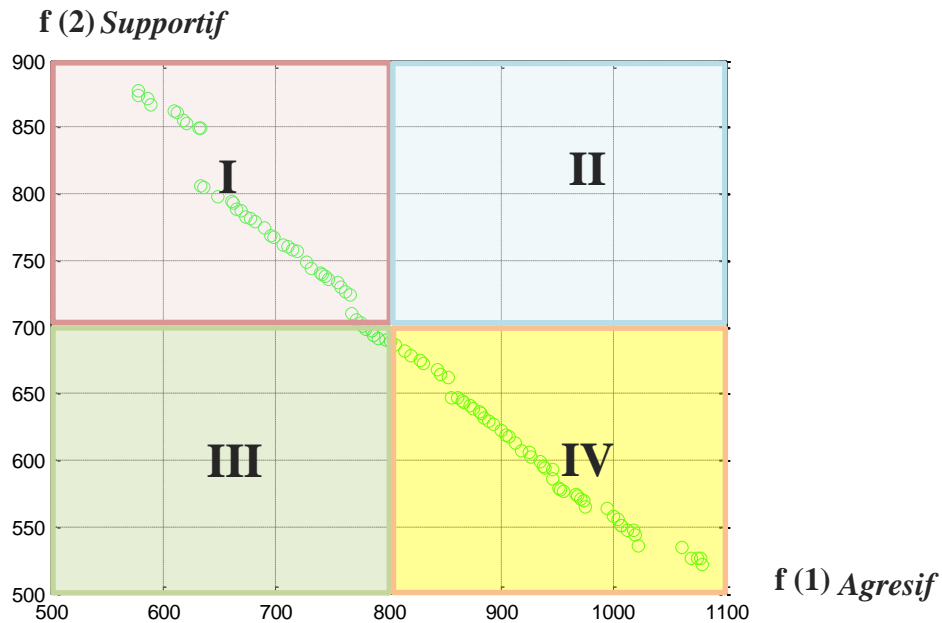
Tabel diatas akan ditunjukkan secara lengkap pada Lampiran 4. Dari beberapa yang diambil dari Tabel 4.4 pada Lampiran 4 didapatkan bahwa seluruh datanya menunjukkan antara dua objective yang berbeda yang merupakan tujuan dari penelitian ini, yaitu antara pahlawan yang tidak hanya dapat menyerang menghancurkan musuhnya namun juga dapat melindungi pasukannya sehingga pasukannya tersebut masih tetap hidup (*supportif*) meskipun ia sendiri harus mati atau mengorbankan nyawanya. Dari keseluruhan data, 100 populasi dengan 20 generasi, didapatkan seluruh populasi menunjukkan Batara yang sangat *agresif*, karena didapatkan f(1) atau perilaku *agresif* lah yang paling banyak muncul (berjumlah 76 dari 100 populasi).

Seluruh data tersebut akan membentuk sebuah *pareto front* antara dua objective yang berbeda atau saling bertentangan tersebut. Dimana grafik *pareto front* disebut pula *pareto optimal* yaitu kumpulan solusi yang paling optimal dari setiap kondisi yang ditemui oleh Batara. Grafik *pareto optimal* ditunjukkan oleh Gambar 4.7.



Gambar 4.7 *Pareto Optimal* Batara dengan 100 Populasi 100 Generasi

Dari grafik diatas didapat bahwa terjadi penyebaran titik pada *pareto front*, dimana titik – titik tersebut adalah sekumpulan solusi dari setiap kondisi yang ada pada 100 populasi. Populasi dapat dikatakan sebagai individu atau dalam *game* dapat kita sebut sebagai skenario. Jadi dari 100 populasi atau 100 skenario pengujian dengan 100 generasi, 100 generasi adalah 100 kali pembangkitan ulang generasi yang lebih baik, sehingga didapatkan kesimpulan dari grafik tersebut bahwa dengan 100 kali pembangkitan ulang generasi yang lebih baik, terjadi penyebaran titik yang juga lebih baik, hampir memenuhi kondisi *pareto front* atau *pareto optimal* dari dua objective yang saling bertentangan. Karena pada generasi 100 titik tersebar lebih baik dari 75 generasi yaitu titik solusi optimal tersebar hampir merata pada kuadran I, III, dan kuadran IV. Penyebaran tersebut agar lebih mudah membacanya, maka akan dibagi menjadi 4 kuadran pula. berikut tampilan grafik yang telah terbagi menjadi 4 kuadran, ditunjukkan oleh Gambar 4.8.

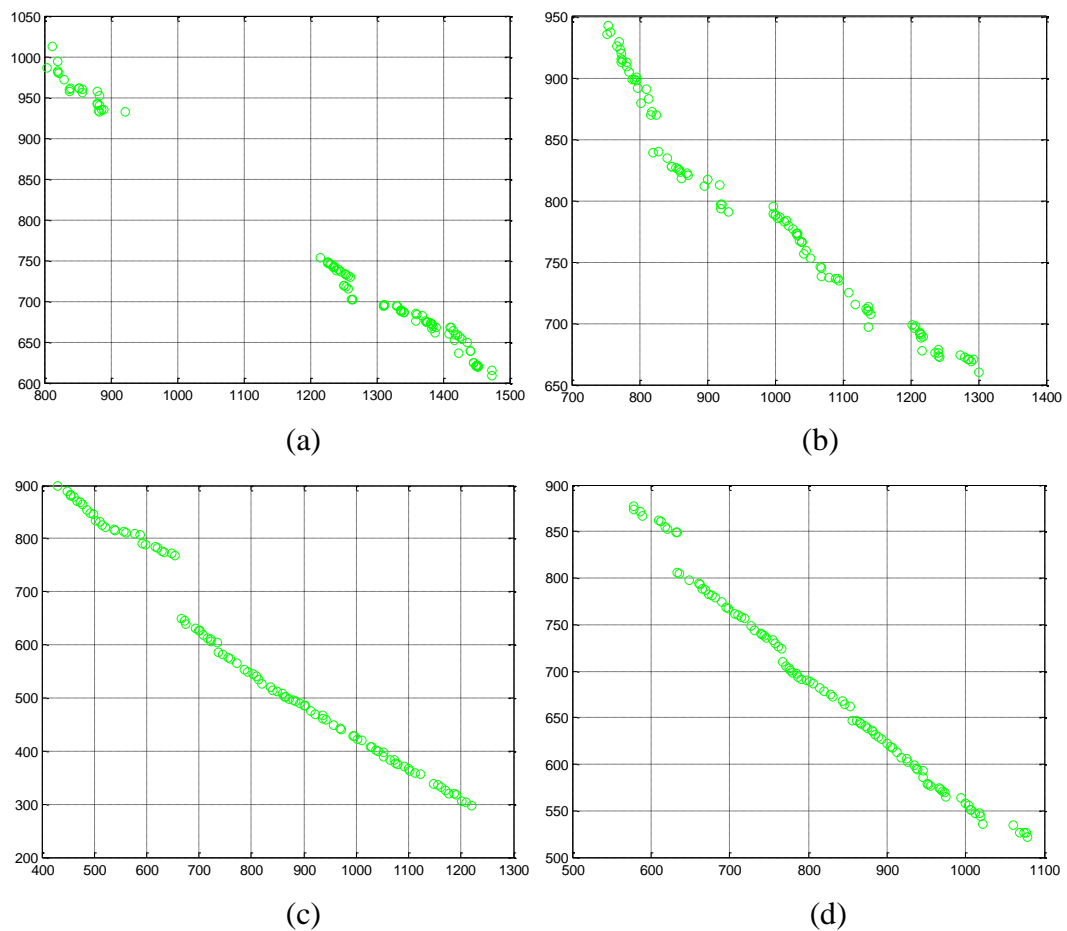


Gambar 4.8 Pembagian 4 kuadran *pareto front* 100 populasi 100 generasi

Maka didapatkan grafik seperti Gambar 4.8 diatas. Grafik tersebut dibagi menjadi 4 kuadran. Kuadran I berwarna merah dimana titik – titik solusi optimal tersebar, menunjukkan perilaku *supportif* yang sangat tinggi namun perilaku *agresif* yang sangat rendah, disini titik – titik solusi optimal tersebut sedikit menyebar mendekati kuadran III yaitu perilaku *supportif* yang sedikit tinggi dan perilaku *agresif* yang sedikit tinggi pula. Kuadran II yang berwarna biru tidak terdapat satupun titik solusi optimal yang tersebar, menunjukkan tidak adanya perilaku *supportif* yang tinggi dan perilaku *agresif* yang tinggi pula untuk pahlawan, karena perilaku yang optimal atau yang diinginkan tidak mungkin tinggi pada keduanya, pasti lebih tinggi salah satu atau berada di tengahnya, itulah solusi yang paling optimal untuk dua tujuan yang saling bertentangan. Untuk kuadran III yang berwarna hijau terdapat beberapa titik solusi optimal didalamnya, menunjukkan perilaku *supportif* yang sedikit tinggi dengan perilaku *agresif* nya yang mendekati nilai tinggi pula. Sedangkan pada kuadran IV yang berwarna kuning inilah titik – titik solusi optimal tersebar lagi yaitu pada perilaku *supportif* yang sangat rendah namun perilaku *agresif* yang sangat tinggi. Dari *pareto optimal* dengan 100 populasi 100 generasi, didapatkan titik – titik solusi optimal hampir rata tersebar pada kuadran I, III, dan IV membentuk garis lurus yang menunjukkan bahwa titik

– titik solusi optimal yang juga membentuk *pareto front* yang hampir sempurna. Sehingga solusi perilaku optimal untuk pahlawan Batara tersebar diantara perilaku yang sangat tinggi *supportif* nya dan sangat rendah *agresif*nya, yang sedikit tinggi *supportif* dan *agresif*nya, serta yang sangat tinggi *agresif*nya namun sangat rendah *supportif* nya.

Dari keempat pengujian 100 populasi yang berbeda generasi, yaitu 20, 50, 75, dan 100 didapat bahwa sedikitnya generasi (yaitu 20 generasi) belum menunjukkan *pareto optimal* untuk dua objective yang berbeda. Penyebarannya belum begitu bagus, sehingga dengan percobaan menambah generasi, pada 100 generasi lah titik solusi optimal yang membentuk *pareto front* tersebar hampir sempurna, atau hampir memenuhi kondisi *pareto optimal*. Berikut perbandingan setiap generasi yang diujikan ditunjukkan oleh Gambar 4.9 :



Gambar 4.9 Perbandingan setiap generasi pada 100 populasi

(a) 20 Generasi (b) 50 Generasi (c) 75 Generasi (d) 100 Generasi

Berdasarkan keempat pengujian untuk dua fungsi objective yang berbeda, dimana dua fungsi objective ini merupakan dua tujuan atau dua perilaku pahlawan yang saling bertentangan, yaitu pahlawan yang diinginkan adalah yang tidak hanya dapat menghancurkan musuhnya dan berhasil menang, namun juga pahlawan yang dapat melindungi pasukannya. Data pengujian pada 100 generasilah yang paling stabil dan dapat membentuk *pareto optimal* dari dua fungsi objective yang saling bertentangan. Sehingga bila terdapat lima perilaku atau lima fungsi objective yang bertentangan pula, maka pengujian lima perilaku ini menggunakan 100 populasi dengan 100 generasi.

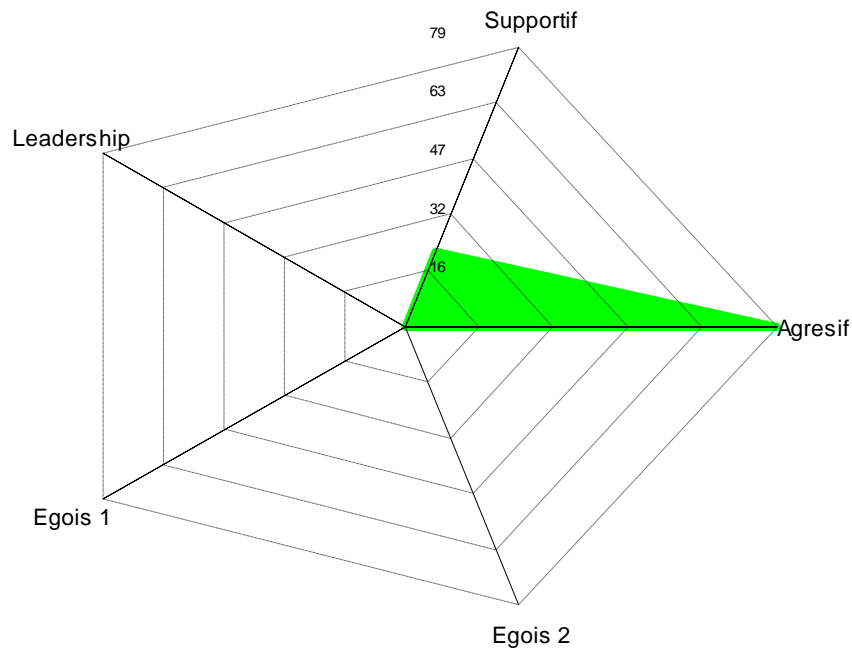
Tabel 4.5 di bawah ini akan menjelaskan bahwa untuk perilaku optimal dari pahlawan Batara Maheswara pada 100 populasi atau 100 skenario dengan 100 generasi, didapatkan perilaku optimal yang sering muncul adalah *agresif* yaitu perilaku pahlawan yang terus menerus menyerang untuk menghancurkan bangunan musuh sebanyak – banyaknya demi memenangkan pertempuran. Sebagai contoh, kita ambil salah satu scenario dari pengujian 100 skenario tersebut, dapat dilihat untuk data ke-97, dimana pada saat kondisi kesehatan pahlawan yaitu $x(1)$ berada pada kondisi paling lemah (pahlawan berada pada level 1), dengan melihat kondisi kesehatan musuh atau $x(2)$ yang berada pada kondisi kesehatan maksimal yaitu bernilai 5778, tidak memungkinkan seorang pahlawan dengan level paling rendah tersebut menyerang sendirian, maka yang perlu dilihat selanjutnya adalah $x(3)$ atau kondisi kesehatan pasukan. Meskipun pada scenario ini pahlawan yang berada pada level yang terendah dengan kesehatan yang sangat rendah pula bila dibandingkan kesehatan musuhnya, dengan memiliki pasukan yang memiliki kesehatan yang besar yaitu bernilai 3900, maka disini keberanian pahlawan muncul, terlebih lagi dengan melihat kondisi selanjutnya yaitu *damage received* pahlawan atau *damage* yang diterima pahlawan $x(4)$ memiliki nilai yang lebih kecil dari *damage* yang akan diterima oleh musuhnya $x(5)$, maka pada kondisi *damage* yang diterima pasukannya juga lebih kecil dari musuhnya pahlawan tersebut pasti akan melakukan atau memilih perilaku *agresif*. Dari keseluruhan hasil pengujian scenario dan mendapatkan beberapa perilaku optimal yang sering muncul, maka dapat ditarik kesimpulan bahwa pahlawan Batara adalah pahlawan yang sangat

agresif karena perilaku optimal yang sering muncul padanya saat kondisi tersebut adalah *agresif*.

Tabel 4.5 Hasil NSGA-II untuk Lima Perilaku Batara dengan 100 populasi dan 100 generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)	f(3)	f(4)	f(5)
82	105.8051	6081.016	163.3828	23.97904	1137.262	234.5291	833.8062	766.1041	-128.014	-1245.84	-2478.41
83	105.9017	6077.947	759.4802	24.58241	1203.661	217.7735	924.7177	725.1487	-42.1095	-1263.23	-2493.89
84	104.9357	6090	2054.355	35.11873	1070.341	284.0521	1005.691	609.6065	192.7152	-1203.18	-2328.42
85	106.6413	6090	70.0636	8.202322	1143.068	243.4554	845.7919	783.7177	-136.939	-1291.56	-2572.67
86	100.0473	6066.786	1833.668	107.088	808.2167	90.90909	781.8606	647.1223	143.504	-939.563	-1814.97
87	100	5784.851	3932.894	57.75085	703.8131	230.9379	1118.399	523.784	587.5718	-952.456	-1766.12
88	104.7905	6080.721	1517.884	41.52912	1042.717	281.7707	935.7259	649.9911	110.5232	-1170.92	-2281.85
89	106.3222	6078.306	2161.905	27.02739	1234.949	245.4934	1107.607	624.3926	193.8642	-1259.35	-2438.46
90	100.0681	5637.964	4753.678	3.869149	618.8366	254.391	1259.62	504.1183	771.7013	-1041.51	-1916.09
91	100.0748	5605.612	4162.115	3.778643	678.2547	218.7996	1220.87	555.6394	665.4386	-1048.04	-1950.05
92	100	5607.468	3459.191	5.313941	634.6681	246.059	1123.957	596.3989	558.721	-1030.29	-1937.07
93	100.0583	6090	1357.729	106.2767	787.5971	76.47837	711.9578	683.4779	61.64498	-944.902	-1842
94	100.1222	6090	1318.314	110.3922	784.1067	88.73554	707.8578	683.2235	58.63499	-929.062	-1811.22
95	105.9619	6083.057	744.6452	25.08493	1198.08	215.3678	921.2469	728.3426	-43.8473	-1260.92	-2489.84
96	104.9999	6089.089	2207.102	35.73523	1071.64	282.6793	1026.184	601.4527	219.092	-1200.22	-2317.44
97	100	5778.11	3900.547	60.71973	702.5531	230.7161	1117.14	527.8519	584.7141	-939.331	-1740.95
98	106.3058	6088.771	1990.808	26.36394	1169.062	229.5453	1066.702	650.7562	175.2999	-1244.21	-2414.41
99	105.7734	6071.073	701.7189	25.12299	1248.215	218.6515	928.1983	719.8361	-61.4155	-1275.34	-2520
100	100.0216	6084.709	649.7993	73.20256	879.0555	90.39457	654.0622	708.2996	-80.8415	-1083.98	-2143.29
Jumlah							78	22	0	0	0

Berikut radar plot yang menunjukkan bahwa Batara adalah pahlawan yang sangat *agresif*.



Gambar 4.10 Grafik radar plot kelima perilaku Batara Maheswara

4.1.5 Pengujian Dua Fungsi Objective Wira dengan 100 Populasi dan 20 Generasi

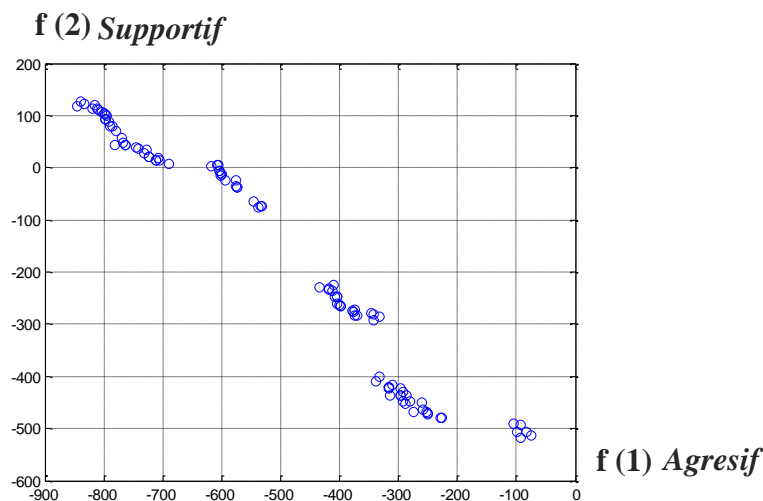
Pada pengujian pertama, untuk Pahlawan Wira Oragastra dengan dua fungsi objective yang berbeda. Berikut Tabel Pengujiannya dengan dua objective yang berbeda (*agresif* dan *supportif*) menggunakan NSGA-II :

Tabel 4.6 Tabel Dua Fungsi Objective Wira 100 Populasi 20 Generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)
96	51.71395	5780.331	5107.391	229.4138	464.928	294.9	-258.775	-463.049
97	50.27334	5469.182	1182.891	189.7477	100.9185	86.33951	-707.748	19.53679
98	55.43761	6021.248	4788.174	55.26153	190.2666	28.51056	-342.684	-280.853
99	52.29291	5471.232	382.0382	69.57582	639.031	169.6303	-605.725	5.886417
100	50.13956	5492.94	5094.659	264.4031	864.0853	265.2486	-83.3439	-506.292
Jumlah							29	71

Tabel diatas akan ditunjukkan secara lengkap pada Lampiran 6. Dari beberapa yang diambil dari Tabel 4.6 pada Lampiran 6 didapatkan bahwa seluruh datanya menunjukkan antara dua objective yang berbeda yang merupakan tujuan dari penelitian ini, yaitu antara pahlawan yang tidak hanya dapat menyerang menghancurkan musuhnya (*agresif*) namun juga dapat melindungi pasukannya sehingga pasukannya tersebut masih tetap hidup (*supportif*) meskipun ia sendiri harus mati atau mengorbankan nyawanya. Dari keseluruhan data, 100 populasi dengan 20 generasi, didapatkan seluruh populasi menunjukkan Wira yang sangat *supportif*, karena didapatkan $f(2)$ atau perilaku *supportif* lah yang paling banyak muncul (berjumlah 71 dari 100 populasi).

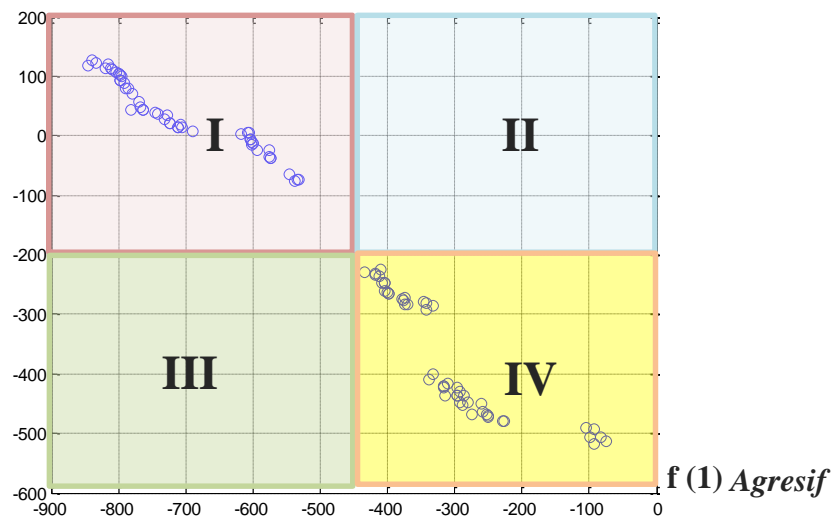
Seluruh data tersebut akan membentuk sebuah *pareto front* antara dua objective yang berbeda atau saling bertentangan tersebut. Dimana grafik *pareto front* disebut pula *pareto optimal* yaitu kumpulan solusi yang paling optimal dari setiap kondisi yang ditemui oleh Wira. Grafik *pareto optimal* ditunjukkan oleh Gambar 4.11.



Gambar 4.11 Pareto Optimal Wira dengan 100 Populasi 20 Generasi

Dari grafik diatas didapat bahwa terjadi penyebaran titik pada *pareto front*, dimana titik – titik tersebut adalah sekumpulan solusi dari setiap kondisi yang ada pada 100 populasi. Populasi dapat dikatakan sebagai individu atau dalam *game* dapat kita sebut sebagai skenario. Jadi dari 100 populasi atau 100 skenario pengujian dengan 20 generasi, 20 generasi adalah 20 kali pembangkitan ulang

generasi yang lebih baik, sehingga didapatkan kesimpulan dari grafik tersebut bahwa solusi optimal tersebar secara terpisah. Penyebaran yang terpisah tersebut agar lebih mudah membacanya, maka akan dibagi menjadi 4 kuadran. berikut tampilan grafik yang telah terbagi menjadi 4 kuadran, ditunjukkan oleh Gambar 4.12.



Gambar 4.12 Pembagian 4 kuadran *pareto front* 100 populasi 20 generasi

Maka didapatkan grafik seperti Gambar 4.2 diatas. Grafik tersebut dibagi menjadi 4 kuadran. Kuadran I berwarna merah dimana titik – titik solusi optimal tersebar, menunjukkan perilaku *supportif* yang sangat tinggi namun perilaku *agresif* yang sangat rendah, sedangkan kuadran II yang berwarna biru tidak terdapat satupun titik solusi optimal yang tersebar, menunjukkan tidak ada perilaku *supportif* yang tinggi dan perilaku *agresif* yang tinggi pula untuk pahlawan. Untuk kuadran III yang berwarna hijau yang tidak ada pula titik solusi optimal yang tersebar, menunjukkan perilaku *supportif* yang rendah dan perilaku *agresif* yang juga rendah bukan perilaku optimal untuk pahlawan. Sedangkan pada kuadran IV yang berwarna kuning inilah titik – titik solusi optimal tersebar kembali yaitu pada perilaku *supportif* yang sangat rendah namun perilaku *agresif* yang sangat tinggi. Dari *pareto optimal* dengan 100 populasi 20 generasi, didapatkan titik – titik solusi optimal paling banyak tersebar pada kuadran I dan IV dimana titik – titik tersebut hampir mendekati bentuk *pareto optimal* pada fungsi minimum untuk perilaku pahlawan Wira.

4.1.6 Pengujian Dua Fungsi Objective Wira dengan 100 Populasi dan 50 Generasi

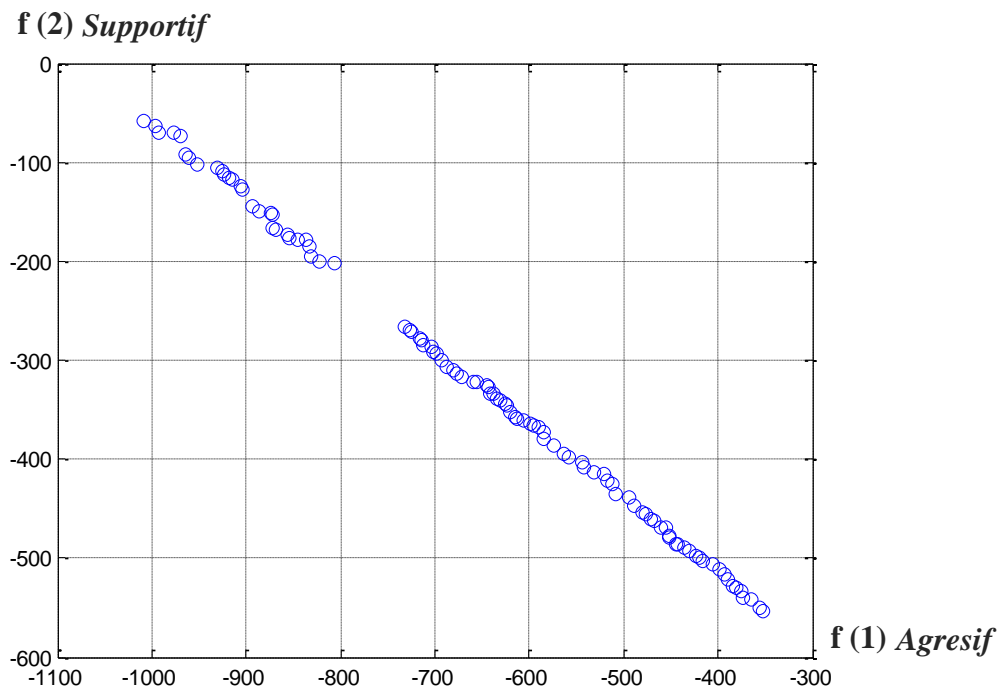
Pada pengujian kedua, untuk Pahlawan Wira Oragatra dengan dua fungsi objective yang berbeda. Berikut Tabel Pengujiannya dengan dua objective yang berbeda (*agresif* dan *supportif*) menggunakan NSGA-II :

Tabel 4.7 Tabel Dua Fungsi Objective Wira 100 Populasi 50 Generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)
96	51.97184	6073.27	472.7157	87.80903	443.7461	154.7683	-914.594	-117.216
97	50	5984.743	4055.274	127.653	600.3053	59.60877	-476.439	-456.323
98	50.01237	5978.784	1963.56	124.0774	572.436	48.21227	-723.988	-272.088
99	50	5976.309	2129.922	122.6121	572.4969	48.87506	-703.835	-286.028
100	50	5979.974	2068.33	124.8499	569.6343	46.32399	-713.314	-280.377
Jumlah							22	78

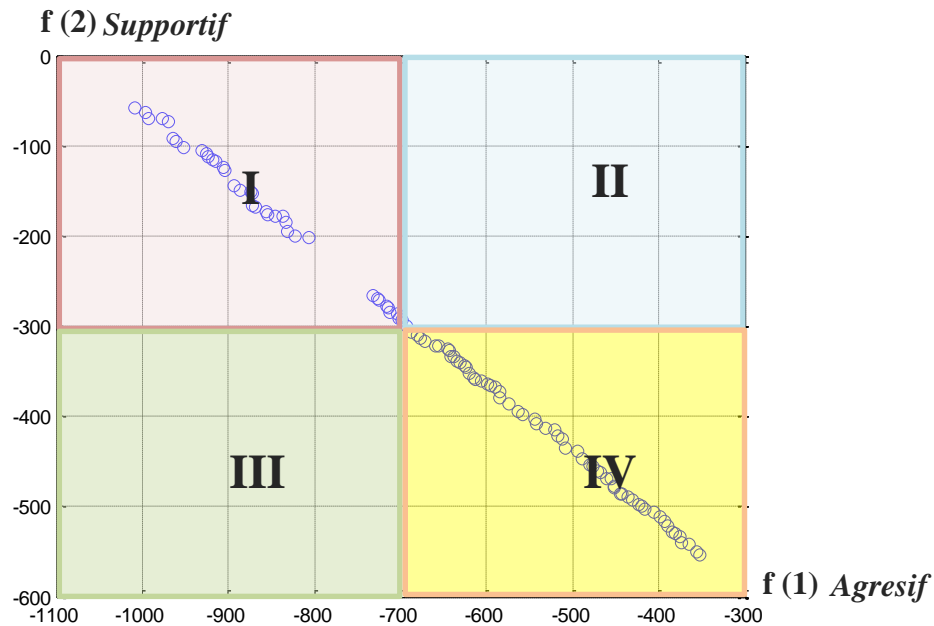
Tabel diatas akan ditunjukkan secara lengkap pada Lampiran 7. Dari beberapa yang diambil dari Tabel 4.7 pada Lampiran 7 didapatkan bahwa seluruh datanya menunjukkan antara dua objective yang berbeda yang merupakan tujuan dari penelitian ini, yaitu antara pahlawan yang tidak hanya dapat menyerang menghancurkan musuhnya (*agresif*) namun juga dapat melindungi pasukannya sehingga pasukannya tersebut masih tetap hidup (*supportif*) meskipun ia sendiri harus mati atau mengorbankan nyawanya. Dari keseluruhan data, 100 populasi dengan 50 generasi, didapatkan seluruh populasi menunjukkan Wira yang sangat *supportif*, karena didapatkan f(2) atau perilaku *supportif* lah yang paling banyak muncul (berjumlah 78 dari 100 populasi).

Seluruh data tersebut akan membentuk sebuah *pareto front* antara dua objective yang berbeda atau saling bertentangan tersebut. Dimana grafik *pareto front* disebut pula *pareto optimal* yaitu kumpulan solusi yang paling optimal dari setiap kondisi yang ditemui oleh Wira. Grafik *pareto optimal* ditunjukkan oleh Gambar 4.3.



Gambar 4.13 *Pareto Optimal* Wira dengan 100 Populasi 50 Generasi

Dari grafik diatas didapat bahwa terjadi penyebaran titik pada *pareto front*, dimana titik – titik tersebut adalah sekumpulan solusi dari setiap kondisi yang ada pada 100 populasi. Populasi dapat dikatakan sebagai individu atau dalam *game* dapat kita sebut sebagai skenario. Jadi dari 100 populasi atau 100 skenario pengujian dengan 50 generasi, 50 generasi adalah 50 kali pembangkitan ulang generasi yang lebih baik, sehingga didapatkan kesimpulan dari grafik tersebut bahwa dengan 50 kali pembangkitan ulang generasi yang lebih baik, terjadi penyebaran titik yang lebih baik pula. Bila pada generasi 20 titik tersebar dengan tingkat kerenggangan antar titik yang besar, dan cenderung terklasifikasi sendiri yaitu hanya pada kuadran I dan kuadran IV, maka pada generasi 50 solusi optimal juga tersebar pada kuadran I dan kuadran IV namun melewati titik tengah yang merupakan perpotongan antar kuadran dan hampir membentuk *pareto front* atau *pareto optimal*. Penyebaran tersebut agar lebih mudah membacanya, maka akan dibagi menjadi 4 kuadran pula. berikut tampilan grafik yang telah terbagi menjadi 4 kuadran, ditunjukkan oleh Gambar 4.14.



Gambar 4.14 Pembagian 4 kuadran *pareto front* 100 populasi 50 generasi

Maka didapatkan grafik seperti Gambar 4.4 diatas. Grafik tersebut dibagi menjadi 4 kuadran. Kuadran I berwarna merah dimana titik – titik solusi optimal tersebar, menunjukkan perilaku *supportif* yang sangat tinggi namun perilaku *agresif* yang sangat rendah. Titik – titik solusi optimal ada beberapa yang tersebar pada kuadran II yang berwarna biru, terdapat beberapa titik solusi optimal yang melewati titik tengah antara kuadran I, II, III, dan IV. Ini menunjukkan terdapat beberapa perilaku *supportif* yang tinggi dan perilaku *agresif* yang tinggi pula untuk pahlawan. Kuadran III tidak tersebar satupun titik solusi optimal didalamnya, menunjukkan perilaku *supportif* yang sangat rendah begitupun perilaku *agresif* nya yang juga sangat rendah bukan perilaku yang optimal untuk kondisi tersebut. Sedangkan pada kuadran IV yang berwarna kuning inilah titik – titik solusi optimal tersebar kembali yaitu pada perilaku *supportif* yang sangat rendah namun perilaku *agresif* yang sangat tinggi. Dari *pareto optimal* dengan 100 populasi 50 generasi, didapatkan titik – titik solusi optimal paling banyak tersebar pada kuadran I dan IV yang menunjukkan bahwa perilaku optimal untuk pahlawan Wira adalah perilaku yang sangat tinggi *supportif* nya dan sangat rendah *agresif* nya serta memiliki pula perilaku pada kuadran IV yatu sangat tinggi *agresif* nya namun sangat rendah

supportif nya. Titik – titik solusi optimal perilaku pahlawan di ketiga kuadran masih belum statbil dalam pembentukan *pareto optimal* untuk fungsi minimum NSGA-II.

4.1.7 Pengujian Dua Fungsi Objective Wira dengan 100 Populasi dan 75 Generasi

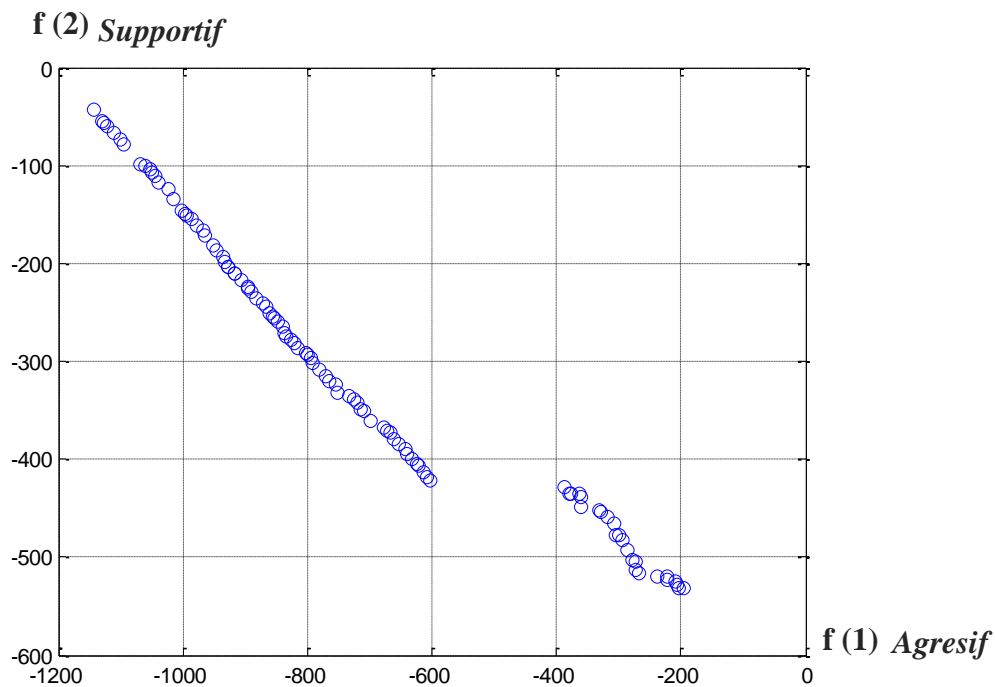
Pada pengujian ketiga, untuk Pahlawan Wira Oragastra dengan dua fungsi objective yang berbeda. Berikut Tabel Pengujiannya dengan dua objective yang berbeda (*agresif* dan *supportif*) menggunakan NSGA-II :

Tabel 4.8 Tabel Dua Fungsi Objective Batara 100 Populasi 75 Generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)
96	50	6066.834	3378.328	219.8576	339.1109	138.5992	-661.994	-378.722
97	50	6090	2153.551	216.2691	187.7256	152.2224	-853.104	-255.936
98	50	6089.648	1833.438	217.3806	191.6839	151.4522	-889.32	-229.66
99	50	6090	1814.485	218.5451	179.3296	151.5938	-894.816	-225.617
100	50	6089.047	2654.98	215.3731	162.333	152.8573	-800.531	-292.654
Jumlah							25	75

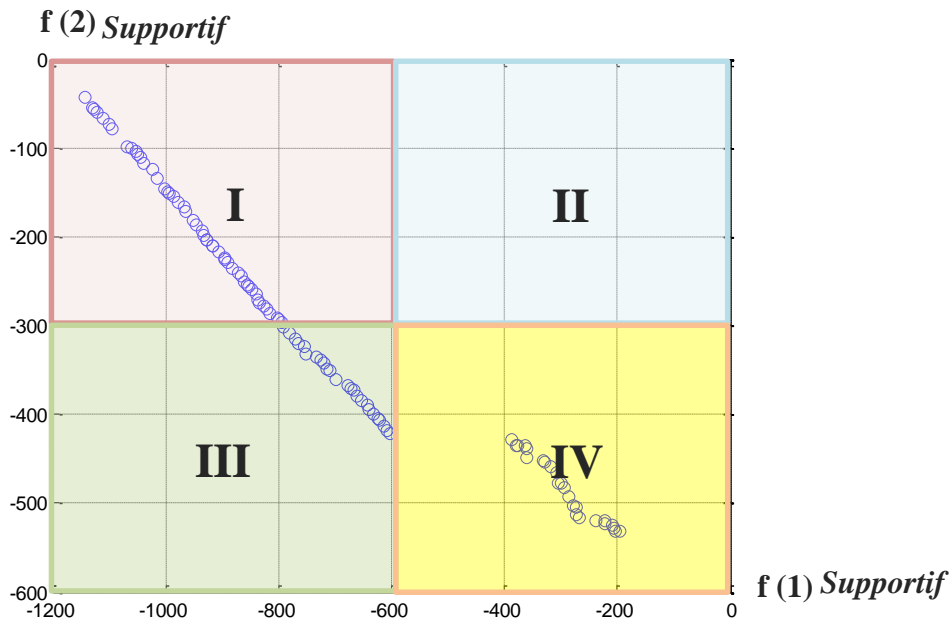
Tabel diatas akan ditunjukkan secara lengkap pada Lampiran 8. Dari beberapa yang diambil dari Tabel 4.8 pada Lampiran 8 didapatkan bahwa seluruh datanya menunjukkan antara dua objective yang berbeda yang merupakan tujuan dari penelitian ini, yaitu antara pahlawan yang tidak hanya dapat menyerang menghancurkan musuhnya namun juga dapat melindungi pasukannya sehingga pasukannya tersebut masih tetap hidup (*supportif*) meskipun ia sendiri harus mati atau mengorbankan nyawanya. Dari keseluruhan data, 100 populasi dengan 100 generasi, didapatkan seluruh populasi menunjukkan Wira yang sangat *supportif*, karena didapatkan f(2) atau perilaku *supportif* lah yang paling banyak muncul (berjumlah 75 dari 100 populasi).

Seluruh data tersebut akan membentuk sebuah *pareto front* antara dua objective yang berbeda atau saling bertentangan tersebut. Dimana grafik *pareto front* disebut pula *pareto optimal* yaitu kumpulan solusi yang paling optimal dari setiap kondisi yang ditemui oleh Wira. Grafik *pareto optimal* ditunjukkan oleh Gambar 4.15.



Gambar 4.15 *Pareto Optimal Wira dengan 100 Populasi 75 Generasi*

Dari grafik diatas didapat bahwa terjadi penyebaran titik pada *pareto front*, dimana titik – titik tersebut adalah sekumpulan solusi dari setiap kondisi yang ada pada 100 populasi. Populasi dapat dikatakan sebagai individu atau dalam *game* dapat kita sebut sebagai skenario. Jadi dari 100 populasi atau 100 skenario pengujian dengan 75 generasi, 75 generasi adalah 75 kali pembangkitan ulang generasi yang lebih baik, sehingga didapatkan kesimpulan dari grafik tersebut bahwa dengan 75 kali pembangkitan ulang generasi yang lebih baik, terjadi penyebaran titik yang juga lebih baik. Bila pada generasi 50 titik tersebar titik solusi optimal membentuk garis lurus *pareto front* atau *pareto optimal* untuk fungsi minimum NSGA-II maka pada generasi 75 titik solusi optimal juga telah membentuk garis lurus *pareto optimal* namun sedikit melengkung. *Pareto optimal* seharusnya membentuk garis lengkung fungsi minimum seperti pada Gambar 2.16. Penyebaran tersebut agar lebih mudah membacanya, maka akan dibagi menjadi 4 kuadran pula. berikut tampilan grafik yang telah terbagi menjadi 4 kuadran, ditunjukkan oleh Gambar 4.16.



Gambar 4.16 Pembagian 4 kuadran *pareto front* 100 populasi 75 generasi

Maka didapatkan grafik seperti Gambar 4.16 diatas. Grafik tersebut dibagi menjadi 4 kuadran. Kuadran I berwarna merah dimana titik – titik solusi optimal tersebar, menunjukkan perilaku *supportif* yang sangat tinggi namun perilaku *agresif* yang sangat rendah, disini titik – titik solusi optimal tersebut menyebar lebih luas hampir mendekati kuadran III yaitu perilaku *supportif* yang sedikit tinggi dan *agresif* yang sedikit tinggi pula. Kuadran II yang berwarna biru tidak terdapat satupun titik solusi optimal yang tersebar, menunjukkan tidak adanya perilaku *supportif* yang tinggi dan perilaku *agresif* yang tinggi pula untuk pahlawan. Untuk kuadran III yang berwarna hijau terdapat beberapa titik solusi optimal didalamnya, menunjukkan perilaku *supportif* yang sedikit tinggi dengan perilaku *agresif* nya yang mendekati nilai tinggi pula. Sedangkan pada kuadran IV yang berwarna kuning inilah titik – titik solusi optimal tersebar lagi yaitu pada perilaku *supportif* yang sangat rendah namun perilaku *agresif* yang sangat tinggi. Dari *pareto optimal* dengan 100 populasi 75 generasi, didapatkan titik – titik solusi optimal hampir rata tersebar pada kuadran I, III, dan IV yang menunjukkan bahwa titik – titik solusi optimal hampir membentuk *pareto front* yang sempurna. Sehingga solusi perilaku optimal untuk pahlawan Wira tersebar diantara perilaku yang sangat tinggi

supportifnya nya dan sangat rendah *agresif* nya, yang sedikit tinggi *supportif* dan *agresif* nya, serta yang sangat tinggi *agresifnya* dan sangat rendah *supportif* nya.

4.1.8 Pengujian Dua Fungsi Objective Wira dengan 100 Populasi dan 100 Generasi

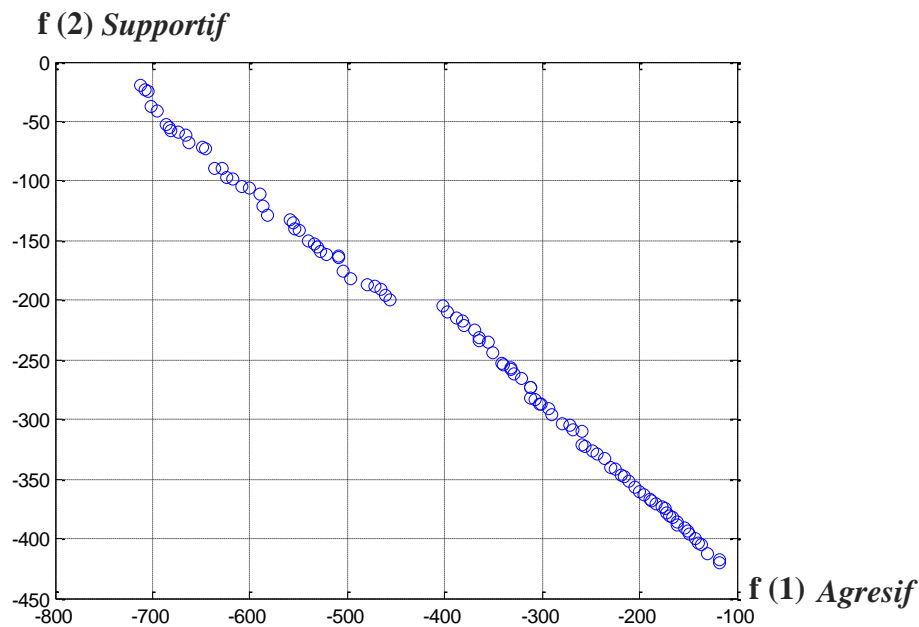
Pada pengujian keempat, untuk Pahlawan Wira Oragastra dengan dua fungsi objective yang berbeda. Berikut Tabel Pengujiannya dengan dua objective yang berbeda (*agresif* dan *supportif*) menggunakan NSGA-II :

Tabel 4.9 Tabel Dua Fungsi Objective Wira 100 Populasi 100 Generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)
96	50	5474.132	4836.026	245.3956	539.9276	62.72901	-183.21	-370.663
97	50.01302	5464.227	3904.557	252.0209	365.4407	75.46256	-331.465	-257.798
98	50.88982	5519.176	1330.417	3.686488	754.5912	176.4435	-527.09	-158.628
99	50.00781	5468.473	4846.146	212.1484	557.6041	56.40539	-173.604	-375.137
100	51.04547	5540.367	35.85008	12.99266	766.8422	195.9782	-680.534	-57.7789
Jumlah							37	63

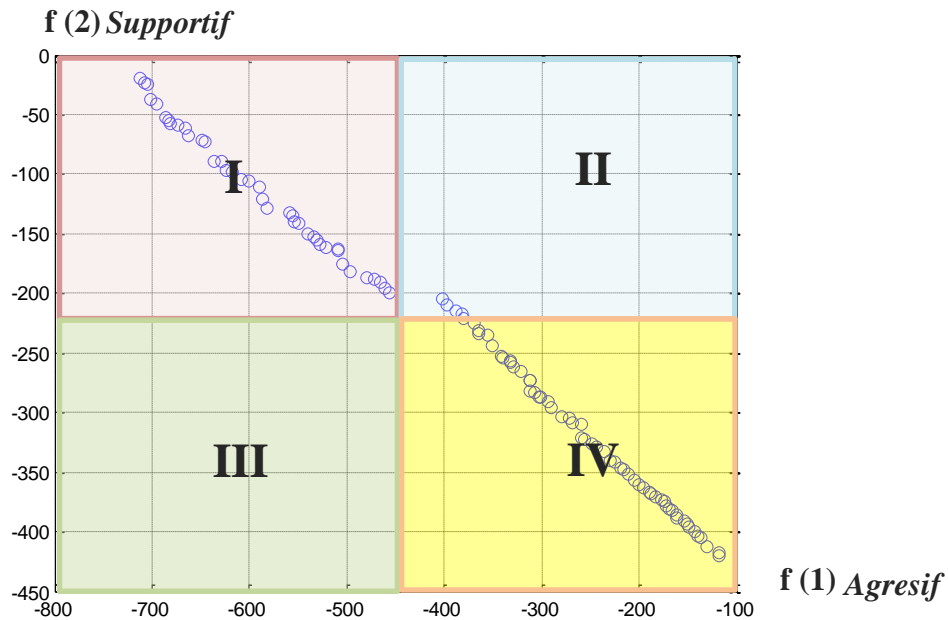
Tabel diatas akan ditunjukkan secara lengkap pada Lampiran 9. Dari beberapa yang diambil dari Tabel 4.9 pada Lampiran 9 didapatkan bahwa seluruh datanya menunjukkan antara dua objective yang berbeda yang merupakan tujuan dari penelitian ini, yaitu antara pahlawan yang tidak hanya dapat menyerang menghancurkan musuhnya namun juga dapat melindungi pasukannya sehingga pasukannya tersebut masih tetap hidup (*supportif*) meskipun ia sendiri harus mati atau mengorbankan nyawanya. Dari keseluruhan data, 100 populasi dengan 20 generasi, didapatkan seluruh populasi menunjukkan Wira yang sangat *supportif*, karena didapatkan f(2) atau perilaku *supportif* lah yang paling banyak muncul (berjumlah 63 dari 100 populasi).

Seluruh data tersebut akan membentuk sebuah *pareto front* antara dua objective yang berbeda atau saling bertentangan tersebut. Dimana grafik *pareto front* disebut pula *pareto optimal* yaitu kumpulan solusi yang paling optimal dari setiap kondisi yang ditemui oleh Wira. Grafik *pareto optimal* ditunjukkan oleh Gambar 4.17.



Gambar 4.17 *Pareto Optimal* Wira dengan 100 Populasi 100 Generasi

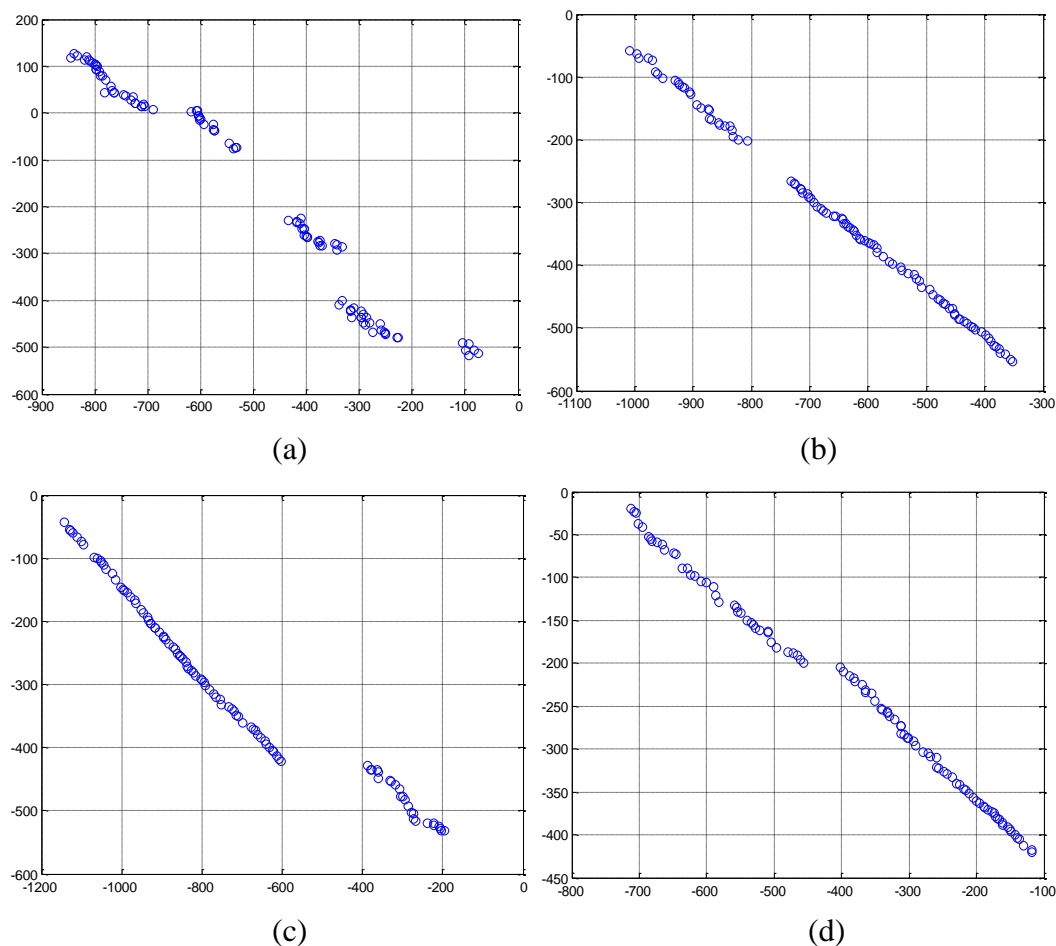
Dari grafik diatas didapat bahwa terjadi penyebaran titik pada *pareto front*, dimana titik – titik tersebut adalah sekumpulan solusi dari setiap kondisi yang ada pada 100 populasi. Populasi dapat dikatakan sebagai individu atau dalam *game* dapat kita sebut sebagai skenario. Jadi dari 100 populasi atau 100 skenario pengujian dengan 100 generasi, 100 generasi adalah 100 kali pembangkitan ulang generasi yang lebih baik, sehingga didapatkan kesimpulan dari grafik tersebut bahwa dengan 100 kali pembangkitan ulang generasi yang lebih baik, terjadi penyebaran titik yang juga lebih baik, hampir memenuhi kondisi *pareto front* atau *pareto optimal* dari dua objective yang saling bertentangan. Karena pada generasi 100 titik tersebar lebih baik dari 75 generasi yaitu titik solusi optimal tersebar merata membentuk garis fungsi minimum NSGA-II yang merupakan *pareto optimal*. Penyebaran tersebut agar lebih mudah membacanya, maka akan dibagi menjadi 4 kuadran pula. berikut tampilan grafik yang telah terbagi menjadi 4 kuadran, ditunjukkan oleh Gambar 4.18 :



Gambar 4.18 Pembagian 4 kuadran *pareto front* 100 populasi 100 generasi

Maka didapatkan grafik seperti Gambar 4.8 diatas. Grafik tersebut dibagi menjadi 4 kuadran. Kuadran I berwarna merah dimana titik – titik solusi optimal tersebar, menunjukkan perilaku *supportif* yang sangat tinggi namun perilaku *agresif* yang sangat rendah, disini titik – titik solusi optimal tersebut sedikit menyebar mendekati kuadran II yaitu perilaku *supportif* yang sedikit tinggi dan perilaku *agresif* yang sedikit tinggi pula. Kuadran III yang berwarna hijau tidak terdapat satupun titik solusi optimal yang tersebar, menunjukkan tidak adanya perilaku *supportif* yang sangat rendah dan perilaku *agresif* yang sangat rendah pula untuk pahlawan. Sedangkan pada kuadran IV yang berwarna kuning inilah titik – titik solusi optimal tersebar lagi yaitu pada perilaku *supportif* yang sangat rendah namun perilaku *agresif* yang sangat tinggi. Dari *pareto optimal* dengan 100 populasi 100 generasi, didapatkan titik – titik solusi optimal hampir rata tersebar pada kuadran I, II, dan IV membentuk garis lurus yang menunjukkan bahwa titik – titik solusi optimal yang juga hampir membentuk *pareto front* fungsi minimum. Sehingga solusi perilaku optimal untuk pahlawan Wira tersebar diantara perilaku yang sangat tinggi *supportif* nya dan sangat rendah *agresif*nya, yang sedikit tinggi *supportif* dan *agresif*nya, serta yang sangat tinggi *agresif*nya namun sangat rendah *supportif*nya.

Dari keempat pengujian 100 populasi yang berbeda generasi, yaitu 20, 50, 75, dan 100 didapat bahwa sedikitnya generasi (yaitu 20 generasi) belum menunjukkan *pareto optimal* untuk dua objective yang berbeda. Penyebarannya belum begitu bagus, sehingga dengan percobaan menambah generasi, pada 100 generasi lah titik solusi optimal yang membentuk *pareto front* tersebar hampir sempurna, atau hampir memenuhi kondisi *pareto optimal*. Berikut perbandingan setiap generasi yang diujikan ditunjukkan oleh Gambar 4.19 :



Gambar 4.19 Perbandingan setiap generasi pada 100 populasi

(a) 20 Generasi (b) 50 Generasi (c) 75 Generasi (d) 100 Generasi

Berdasarkan keempat pengujian untuk dua fungsi objective yang berbeda, dimana dua fungsi objective ini merupakan dua tujuan atau dua perilaku pahlawan yang saling bertentangan, yaitu pahlawan yang diinginkan adalah yang tidak hanya dapat menghancurkan musuhnya dan berhasil menang, namun juga pahlawan

yang dapat melindungi pasukannya. Data pengujian pada 100 generasilah yang paling stabil dan dapat membentuk *pareto optimal* dari dua fungsi objective yang saling bertentangan. Sehingga bila terdapat lima perilaku atau lima fungsi objective yang bertentangan pula, maka pengujian lima perilaku ini menggunakan 100 populasi dengan 100 generasi.

Tabel 4.10 menjelaskan bahwa untuk perilaku optimal dari pahlawan Wira Oragastra pada 100 populasi atau 100 skenario dengan 100 generasi, didapatkan perilaku optimal yang sering muncul adalah *supportif* yaitu perilaku pahlawan yang melindungi pasukannya agar pasukannya tetap hidup meskipun ia sendiri mati. Pada pengujian lima perilaku dengan 100 populasi 100 generasi ini juga membangkitkan perilaku optimal lainnya dari Wira Oragastra yaitu perilaku *agresif* dan *leadership*, hanya saja tidak sebanyak perilaku *supportif* saat dihadapkan kondisi atau scenario tersebut. Sebagai contoh, untuk perilaku *agresif* kita ambil salah satu scenario dari pengujian 100 skenario tersebut, dapat dilihat untuk data ke-96, dimana pada saat kondisi kesehatan pahlawan yaitu $x(1)$ berada pada kondisi paling lemah (pahlawan berada pada level 1), dengan melihat kondisi kesehatan musuh atau $x(2)$ yang berada pada kondisi kesehatan maksimal yaitu bernilai 5792, tidak memungkinkan seorang pahlawan dengan level paling rendah tersebut menyerang sendirian, maka yang perlu dilihat selanjutnya adalah $x(3)$ atau kondisi kesehatan pasukan. Meskipun pada scenario ini pahlawan yang berada pada level yang terendah dengan kesehatan yang sangat rendah pula bila dibandingkan kesehatan musuhnya, dengan memiliki pasukan yang memiliki kesehatan yang besar yaitu bernilai 4071, maka disini keberanian pahlawan muncul, terlebih lagi dengan melihat kondisi selanjutnya yaitu *damage received* pahlawan atau *damage* yang diterima pahlawan $x(4)$ memiliki nilai yang lebih kecil dari *damage* yang akan diterima oleh musuhnya $x(5)$, maka pada kondisi *damage* yang diterima pasukannya juga lebih kecil dari musuhnya pahlawan tersebut pasti akan melakukan atau memilih perilaku *agresif*.

Untuk pengujian scenario yang memunculkan perilaku optimal *supportif* dari pahlawan dapat kita lihat pada skenario ke – 97. Dimana kesehatan pahlawan yang sedikit lebih besar dari kondisi minimum kesehatannya, dengan melihat kondisi kesehatan musuhnya $x(2)$ yang besar, pahlawan mulai melihat kondisi

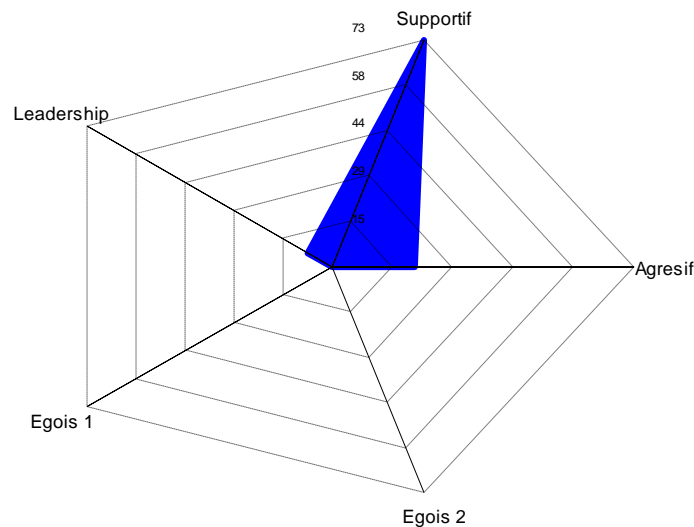
kesehatan pasukannya x(3) yang ternyata juga sangat lemah. Dari kondisi tersebut pastinya pahlawan tidak akan menyerang dan tak mungkin melindungi pasukannya, namun dengan melihat kondisi *damage* yang diterima oleh pahlawan lebih kecil dari musuhnya, maka mulai muncul perilakunya yang ingin melindungi pasukannya meskipun ia sendiri harus mempertaruhkan kehilangan nyawanya. Begitupun dengan pengujian perilaku optimal lain yang muncul pada pahlawan yaitu perilaku *leadership*.

Dapat dilihat pada scenario ke – 100 dimana kesehatannya yang lemah yaitu hanya bernilai 53, ia harus menghadapi musuh dengan kekuatan maksimal yaitu 6090, maka ia akan melihat kondisi pasukannya, pasukannya memiliki nyawa yang cukup besar yaitu 4723. Maka dengan melihat kondisi lain yaitu *damage* yang diterima olehnya dan pasukannya yang lebih rendah dari musuhnya, maka pahlawan akan memilih perilaku *leadership*, yaitu perilaku memimpin, ia akan memimpin pasukannya untuk menyerang bersama – sama dengannya sehingga meskipun ia memiliki nyawa yang lemah, dengan memiliki pasukan yang besar untuk menyerang musuh yang sangat kuat, maka ia akan berperilaku *leadership*. Dari keseluruhan hasil pengujian scenario dan mendapatkan beberapa perilaku optimal yang sering muncul, maka dapat ditarik kesimpulan bahwa pahlawan Wira adalah pahlawan yang sangat *supportif* karena perilaku optimal yang sering muncul padanya saat kondisi tersebut adalah *supportif* meskipun muncul perilaku – perilaku optimal lainnya pada kondisi tertentu.

Tabel 4.10 Hasil NSGA-II untuk Lima Perilaku Wira dengan 100 populasi dan 100 generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)	f(3)	f(4)	f(5)
81	56.82876	6088.71	3109.491	51.18403	591.2987	96.37058	-427.898	-213.673	-498.181	-1377.6	-1760.73
82	56.85082	6064.164	270.4337	101.1395	209.8317	172.0716	-847.378	92.07704	-598.252	-1105.05	-1179.6
83	55.95771	6090	3566.725	49.00591	601.4587	95.04603	-398.588	-280.148	-470.763	-1405.36	-1786.18
84	53.7092	6090	4757.798	57.44912	493.9885	109.6515	-354.721	-428.173	-340.47	-1395.32	-1648.9
85	56.36954	6069.626	30	96.49562	68.29842	162.9177	-927.157	126.4403	-589.076	-1097.04	-1145.73
86	53.46309	6086.21	4791.774	60.13006	466.7692	99.505	-363.054	-429.191	-332.785	-1382.92	-1624.92
87	53.52064	6077.048	4908.425	59.71692	466.7154	99.76719	-343.756	-434.991	-318.067	-1379.31	-1617.76
88	50.03771	5615.723	2900.072	4.179206	895.0923	124.8874	-374.536	-354.295	-618.003	-1548.97	-2154.42
89	50.04605	5721.042	3620.885	9.565405	788.7469	127.9997	-362.699	-419.412	-532.948	-1548.15	-2072.97
90	50.0375	5616.098	2961.687	5.415271	895.1343	120.4133	-367.398	-358.296	-612.826	-1545.02	-2148.82
91	54.06253	6090	4105.675	57.70584	518.0226	117.7233	-414.471	-370.028	-405.575	-1393.4	-1669.81
92	54.71087	6058.358	31.52327	107.2455	58.2024	176.9329	-975.207	78.96058	-597.575	-1087.16	-1080.81
93	54.8478	6060.66	118.4984	98.87189	12	178.2906	-973.266	83.30853	-575.732	-1101.75	-1099.99
94	50	5551.069	739.9899	70.33259	894.9321	203.3104	-604.519	-172.214	-759.142	-1302.24	-1675.3
95	50.00457	5806.28	4784.452	3.466755	899.9404	120.2741	-236.871	-559.847	-481.212	-1632.62	-2200.14
96	50	5792.393	4071.684	3.168	900	105.9957	-313.646	-493.587	-554.014	-1627.94	-2233.36
97	56.82582	6062.36	113.4735	103.8284	193.1313	170.1855	-869.861	108.9386	-607.59	-1091.66	-1157.26
98	50	5547.85	1018.25	72.12167	900	201.4508	-569.392	-194.967	-731.926	-1296.2	-1657.72
99	56.25273	6089.647	3242.623	60.87769	631.7654	107.8719	-420.643	-252.282	-498.551	-1367.31	-1719.14
100	53.80832	6090	4723.208	57.48866	485.8804	110.3669	-357.834	-420.87	-339.481	-1391.17	-1641.69
Jumlah							20	73	7	0	0

Berikut radar plot yang menunjukkan bahwa Wira adalah pahlawan yang sangat *supportif*.



Gambar 4.20 Grafik radar plot kelima perilaku Wira Oragastra

4.1.9 Pengujian Dua Fungsi Objective Kirna dengan 100 Populasi dan 20 Generasi

Pada pengujian pertama, untuk Pahlawan Kirna Waranggani dengan dua fungsi objective yang berbeda. Berikut Tabel Pengujiannya dengan dua objective yang berbeda (*agresif* dan *supportif*) menggunakan NSGA-II :

Tabel 4.11 Tabel Dua Fungsi Objective Kirna 100 Populasi 20 Generasi

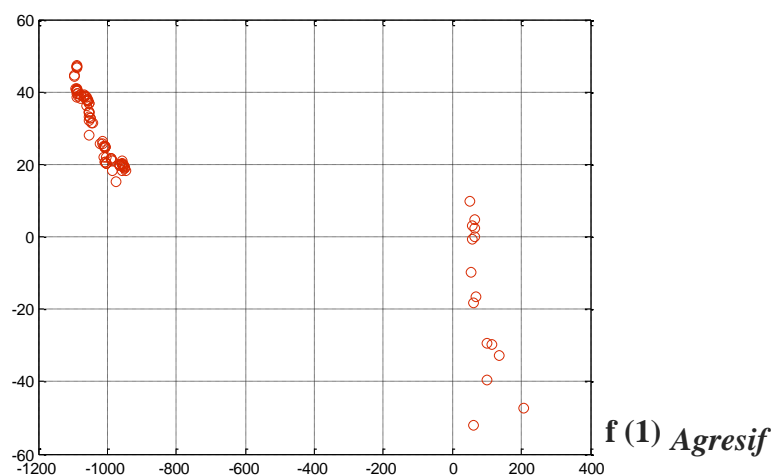
Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)
96	53.51206	6072.836	1892.351	284.8821	131.9168	5.928752	-954.567	19.85013
97	53.62085	6088.38	1758.677	285.2108	30.31886	19.67755	-1087.44	40.36435
98	53.68479	6090	1825.948	285.5946	14.00484	4.006404	-1089.44	47.10894
99	53.62074	6088.569	1754.706	285.2337	30.42823	19.54268	-1088.2	40.4623
100	53.50914	6071.812	1892.458	284.808	132.2663	6.703699	-953.84	19.73217
Jumlah							15	85

Tabel diatas akan ditunjukkan secara lengkap pada Lampiran 11. Dari beberapa yang diambil dari Tabel 4.11 pada Lampiran 11 didapatkan bahwa seluruh datanya menunjukkan antara dua objective yang berbeda yang merupakan tujuan

dari penelitian ini, yaitu antara pahlawan yang tidak hanya dapat menyerang menghancurkan musuhnya (*agresif*) namun juga dapat melindungi pasukannya sehingga pasukannya tersebut masih tetap hidup (*supportif*) meskipun ia sendiri harus mati atau mengorbankan nyawanya. Dari keseluruhan data, 100 populasi dengan 20 generasi, didapatkan seluruh populasi menunjukkan Kirna yang sangat *supportif*, karena didapatkan $f(2)$ atau perilaku *supportif* lah yang paling banyak muncul (berjumlah 85 dari 100 populasi).

Seluruh data tersebut akan membentuk sebuah *pareto front* antara dua objective yang berbeda atau saling bertentangan tersebut. Dimana grafik *pareto front* disebut pula *pareto optimal* yaitu kumpulan solusi yang paling optimal dari setiap kondisi yang ditemui oleh Kirna. Grafik *pareto optimal* ditunjukkan oleh Gambar 4.21.

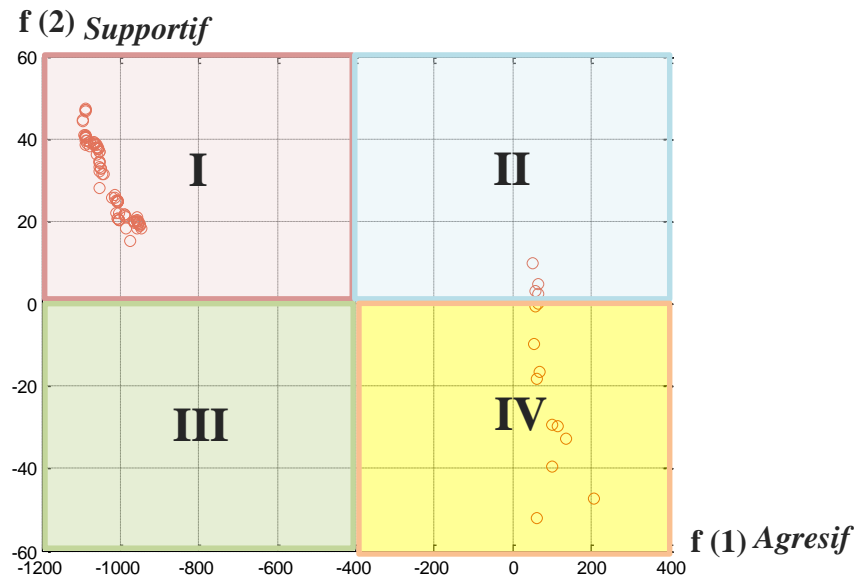
f (2) Supportif



Gambar 4.21 *Pareto Optimal* Kirna dengan 100 Populasi 20 Generasi

Dari grafik diatas didapat bahwa terjadi penyebaran titik pada *pareto front*, dimana titik – titik tersebut adalah sekumpulan solusi dari setiap kondisi yang ada pada 100 populasi. Populasi dapat dikatakan sebagai individu atau dalam *game* dapat kita sebut sebagai skenario. Jadi dari 100 populasi atau 100 skenario pengujian dengan 20 generasi, 20 generasi adalah 20 kali pembangkitan ulang generasi yang lebih baik, sehingga didapatkan kesimpulan dari grafik tersebut bahwa solusi optimal tersebar secara terpisah. Penyebaran yang terpisah tersebut agar lebih mudah membacanya, maka akan dibagi menjadi 4 kuadran. berikut

tampilan grafik yang telah terbagi menjadi 4 kuadran, ditunjukkan oleh Gambar 4.12.



Gambar 4.22 Pembagian 4 kuadran *pareto front* 100 populasi 20 generasi

Maka didapatkan grafik seperti Gambar 4.22 diatas. Grafik tersebut dibagi menjadi 4 kuadran. Kuadran I berwarna merah dimana titik – titik solusi optimal tersebar, menunjukkan perilaku *supportif* yang sangat tinggi namun perilaku *agresif* yang sangat rendah, sedangkan kuadran II yang berwarna biru terdapat beberapa titik solusi optimal yang tersebar, menunjukkan adanya perilaku *supportif* yang sedikit tinggi dan perilaku *agresif* yang tinggi untuk pahlawan. Untuk kuadran III yang berwarna hijau yang tidak ada titik solusi optimal yang tersebar, menunjukkan perilaku *supportif* yang sangat rendah dan perilaku *agresif* yang juga sangat rendah bukan perilaku optimal untuk pahlawan. Sedangkan pada kuadran IV yang berwarna kuning titik – titik solusi optimal tersebar kembali yaitu pada perilaku *supportif* yang sangat rendah namun perilaku *agresif* yang sangat tinggi. Dari *pareto optimal* dengan 100 populasi 20 generasi, didapatkan titik – titik solusi optimal paling banyak tersebar pada kuadran I dan IV dimana titik – titik tersebut belum mendekati bentuk *pareto optimal* pada fungsi minimum NSGA-II pada Kirna.

4.1.10 Pengujian Dua Fungsi Objective Wira dengan 100 Populasi dan 50 Generasi

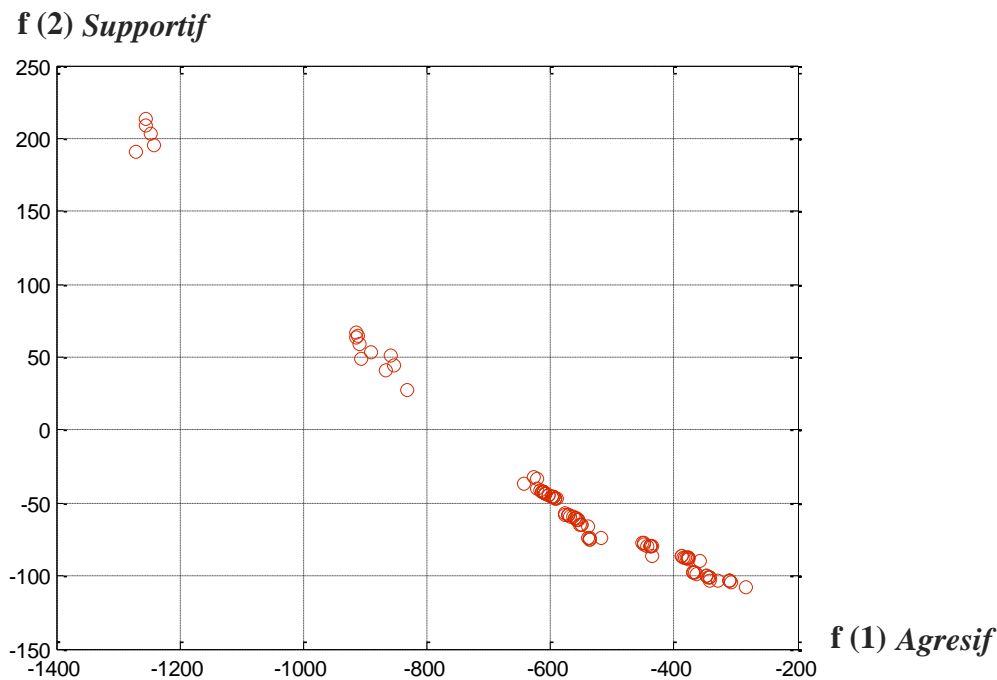
Pada pengujian kedua, untuk Pahlawan Kirna Waranggani dengan dua fungsi objective yang berbeda. Berikut Tabel Pengujiannya dengan dua objective yang berbeda (*agresif* dan *supportif*) menggunakan NSGA-II :

Tabel 4.12 Tabel Dua Fungsi Objective Kirna 100 Populasi 50 Generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)
96	50.00588	5649.35	2030.977	275.8369	418.7748	125.6832	-535.5	-74.1444
97	55.26362	5706.676	50.37488	205.6038	14	131.6866	-1254.06	208.7776
98	50.00003	5603.144	1946.754	282.4972	340.0512	121.4899	-611.974	-42.543
99	50	5624.457	2031.161	275.5847	385.0893	122.4848	-558.837	-60.5345
100	50.03929	5629.709	2036.782	275.6603	394.3752	129.2168	-549.942	-64.5024
Jumlah							0	100

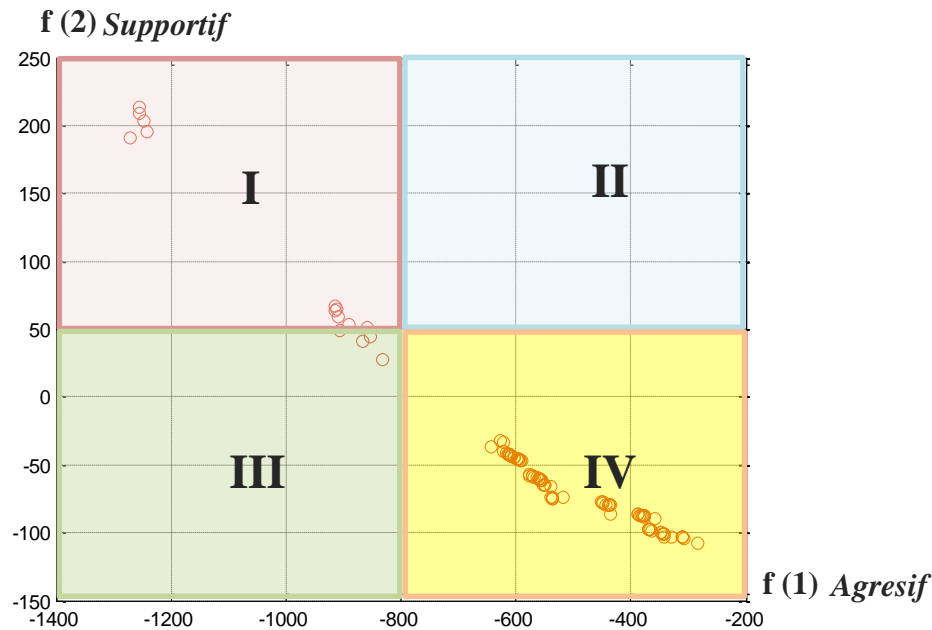
Tabel diatas akan ditunjukkan secara lengkap pada Lampiran 12. Dari beberapa yang diambil dari Tabel 4.12 pada Lampiran 12 didapatkan bahwa seluruh datanya menunjukkan antara dua objective yang berbeda yang merupakan tujuan dari penelitian ini, yaitu antara pahlawan yang tidak hanya dapat menyerang menghancurkan musuhnya (*agresif*) namun juga dapat melindungi pasukannya sehingga pasukannya tersebut masih tetap hidup (*supportif*) meskipun ia sendiri harus mati atau mengorbankan nyawanya. Dari keseluruhan data, 100 populasi dengan 50 generasi, didapatkan seluruh populasi menunjukkan Kirna yang sangat *supportif*, karena didapatkan f(2) atau perilaku *supportif* lah yang paling banyak muncul (berjumlah 100 dari 100 populasi).

Seluruh data tersebut akan membentuk sebuah *pareto front* antara dua objective yang berbeda atau saling bertentangan tersebut. Dimana grafik *pareto front* disebut pula *pareto optimal* yaitu kumpulan solusi yang paling optimal dari setiap kondisi yang ditemui oleh Kirna. Grafik *pareto optimal* ditunjukkan oleh Gambar 4.3.



Gambar 4.23 *Pareto Optimal* Kirna dengan 100 Populasi 50 Generasi

Dari grafik diatas didapat bahwa terjadi penyebaran titik pada *pareto front*, dimana titik – titik tersebut adalah sekumpulan solusi dari setiap kondisi yang ada pada 100 populasi. Populasi dapat dikatakan sebagai individu atau dalam *game* dapat kita sebut sebagai skenario. Jadi dari 100 populasi atau 100 skenario pengujian dengan 50 generasi, 50 generasi adalah 50 kali pembangkitan ulang generasi yang lebih baik, sehingga didapatkan kesimpulan dari grafik tersebut bahwa dengan 50 kali pembangkitan ulang generasi yang lebih baik, terjadi penyebaran titik yang lebih baik pula. Bila pada generasi 20 titik tersebar dengan tingkat kerenggangan antar titik yang besar, dan cenderung terklasifikasi sendiri yaitu hanya pada kuadran I dan kuadran IV, maka pada generasi 50 solusi optimal juga tersebar pada kuadran I dan kuadran IV namun melewati titik tengah yang merupakan perpotongan antar kuadran dan hampir membentuk *pareto front* atau *pareto optimal*. Penyebaran tersebut agar lebih mudah membacanya, maka akan dibagi menjadi 4 kuadran pula. berikut tampilan grafik yang telah terbagi menjadi 4 kuadran, ditunjukkan oleh Gambar 4.14.



Gambar 4.24 Pembagian 4 kuadran *pareto front* 100 populasi 50 generasi

Maka didapatkan grafik seperti Gambar 4.24 diatas. Grafik tersebut dibagi menjadi 4 kuadran. Kuadran I berwarna merah dimana titik – titik solusi optimal tersebar, menunjukkan perilaku *supportif* yang sangat tinggi namun perilaku *agresif* yang sangat rendah. Titik – titik solusi optimal ada beberapa yang tersebar pada kuadran III yang berwarna biru, yang menandakan perilaku optimal pahlawan ada yang *supportifnya* sedikit tinggi, begitupun dengan *agresif* nya yang sedikit tinggi pula. Kuadran II tidak tersebar satupun titik solusi optimal didalamnya, menunjukkan perilaku *suportif* yang sangat tinggi begitupun perilaku *agresif* nya yang juga sangat tinggi bukan perilaku yang optimal untuk kondisi tersebut. Sedangkan pada kuadran IV yang berwarna kuning inilah titik – titik solusi optimal tersebar kembali yaitu pada perilaku *supportif* yang sangat rendah namun perilaku *agresif* yang sangat tinggi. Dari *pareto optimal* dengan 100 populasi 50 generasi, didapatkan titik – titik solusi optimal paling banyak tersebar pada kuadran IV yang menunjukkan bahwa perilaku optimal untuk pahlawan Kirna adalah perilaku yang sangat tinggi *agresif* nya dan sangat rendah *supportif* nya serta memiliki pula sedikit perilaku optimal pada kuadran I yatu sangat tinggi *agresif* nya namun sangat rendah *supportif* nya. Titik – titik solusi optimal perilaku pahlawan di ketiga kuadran masih belum statbil dalam pembentukan *pareto optimal* untuk fungsi minimum NSGA-II.

4.1.11 Pengujian Dua Fungsi Objective Kirna dengan 100 Populasi dan 75 Generasi

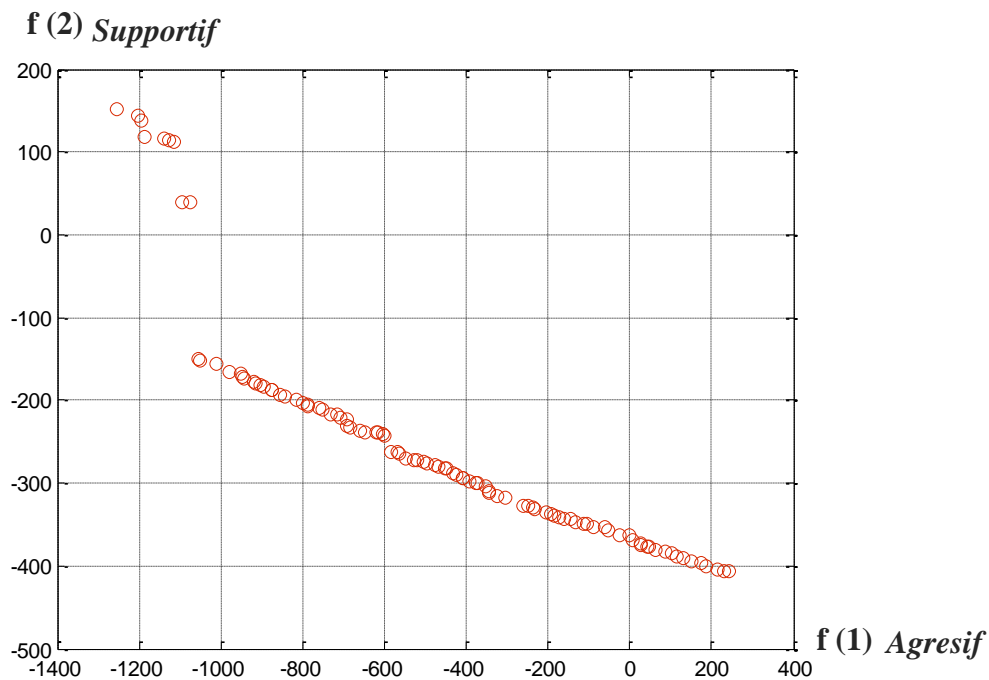
Pada pengujian ketiga, untuk Pahlawan Kirna Warangani dengan dua fungsi objective yang berbeda. Berikut Tabel Pengujiannya dengan dua objective yang berbeda (*agresif* dan *supportif*) menggunakan NSGA-II :

Tabel 4.13 Tabel Dua Fungsi Objective Batara 100 Populasi 75 Generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)
96	50	6090	3963.759	119.3025	607.4087	291.353	-144.184	-343.859
97	50	6089.122	1254.69	106.0983	674.8618	184.5914	-616.179	-239.459
98	50	6088.511	715.6964	113.0899	598.6428	174.5678	-799.892	-202.84
99	50	6088.538	3159.368	132.9766	537.472	285.3122	-374.517	-299.943
100	50.00119	6089.599	3763.183	135.8096	740.2008	268.3433	-51.2077	-356.084
Jumlah							35	65

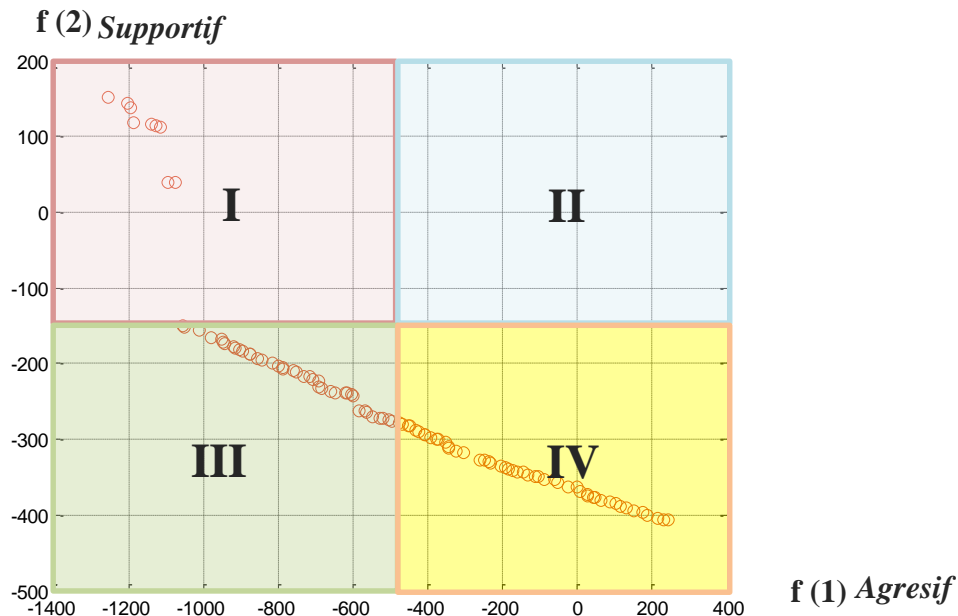
Tabel diatas akan ditunjukkan secara lengkap pada Lampiran 13. Dari beberapa yang diambil dari Tabel 4.13 pada Lampiran 13 didapatkan bahwa seluruh datanya menunjukkan antara dua objective yang berbeda yang merupakan tujuan dari penelitian ini, yaitu antara pahlawan yang tidak hanya dapat menyerang menghancurkan musuhnya namun juga dapat melindungi pasukannya sehingga pasukannya tersebut masih tetap hidup (*supportif*) meskipun ia sendiri harus mati atau mengorbankan nyawanya. Dari keseluruhan data, 100 populasi dengan 100 generasi, didapatkan seluruh populasi menunjukkan Wira yang sangat *supportif*, karena didapatkan f(2) atau perilaku *supportif* lah yang paling banyak muncul (berjumlah 65 dari 100 populasi).

Seluruh data tersebut akan membentuk sebuah *pareto front* antara dua objective yang berbeda atau saling bertentangan tersebut. Dimana grafik *pareto front* disebut pula *pareto optimal* yaitu kumpulan solusi yang paling optimal dari setiap kondisi yang ditemui oleh Kirna. Grafik *pareto optimal* ditunjukkan oleh Gambar 4.15.



Gambar 4.25 *Pareto Optimal Kirna dengan 100 Populasi 75 Generasi*

Dari grafik diatas didapat bahwa terjadi penyebaran titik pada *pareto front*, dimana titik – titik tersebut adalah sekumpulan solusi dari setiap kondisi yang ada pada 100 populasi. Populasi dapat dikatakan sebagai individu atau dalam *game* dapat kita sebut sebagai skenario. Jadi dari 100 populasi atau 100 skenario pengujian dengan 75 generasi, 75 generasi adalah 75 kali pembangkitan ulang generasi yang lebih baik, sehingga didapatkan kesimpulan dari grafik tersebut bahwa dengan 75 kali pembangkitan ulang generasi yang lebih baik, terjadi penyebaran titik yang juga lebih baik. Bila pada generasi 50 titik - titik solusi optimal yang tersebar masih saling berjauhan jaraknya antar titik satu dengan titik lainnya, dan belum membentuk *pareto front* atau *pareto optimal* untuk fungsi minimum NSGA-II, maka pada generasi 75 titik - titik solusi optimal saling berdekatan jarak antar titiknya sehingga membentuk garis lurus mendekati bentuk *pareto optimal*. *Pareto optimal* seharusnya membentuk garis lengkung fungsi minimum seperti pada Gambar 2.16. Penyebaran tersebut agar lebih mudah membacanya, maka akan dibagi menjadi 4 kuadran pula. berikut tampilan grafik yang telah terbagi menjadi 4 kuadran, ditunjukkan oleh Gambar 4.26.



Gambar 4.26 Pembagian 4 kuadran *pareto front* 100 populasi 75 generasi

Maka didapatkan grafik seperti Gambar 4.26 diatas. Grafik tersebut dibagi menjadi 4 kuadran. Kuadran I berwarna merah dimana terdapat beberapa titik solusi optimal tersebar, menunjukkan perilaku *supportif* yang sangat tinggi namun perilaku *agresif* yang sangat rendah. Titik – titik solusi optimal tersebar juga pada kuadran III yaitu perilaku *supportif* yang sedikit tinggi dan *agresif* yang sedikit tinggi pula. Kuadran II yang berwarna biru tidak terdapat satupun titik solusi optimal yang tersebar, menunjukkan tidak adanya perilaku *supportif* yang tinggi dan perilaku *agresif* yang tinggi pula untuk pahlawan. Sedangkan pada kuadran IV yang berwarna kuning, titik – titik solusi optimal tersebar lagi yaitu pada perilaku *supportif* yang sangat rendah namun perilaku *agresif* yang sangat tinggi. Dari *pareto optimal* dengan 100 populasi 75 generasi, didapatkan titik – titik solusi optimal hampir rata tersebar pada kuadran I, III, dan IV yang menunjukkan bahwa titik – titik solusi optimal hampir membentuk *pareto front*. Sehingga solusi perilaku optimal untuk pahlawan Kirna tersebar diantara perilaku yang sangat tinggi *supportifnya* nya dan sangat rendah *agresif* nya, yang sedikit tinggi *supportif* dan *agresif* nya, serta yang sangat tinggi *agresifnya* dan sangat rendah *supportif* nya.

4.1.12 Pengujian Dua Fungsi Objective Wira dengan 100 Populasi dan 100 Generasi

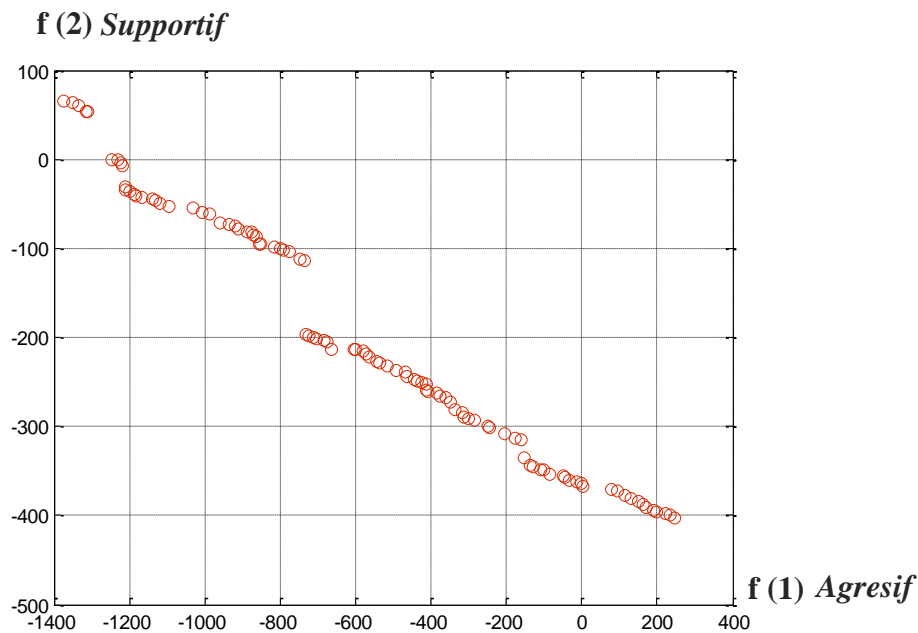
Pada pengujian keempat, untuk Pahlawan Kirna dengan dua fungsi objective yang berbeda. Berikut Tabel Pengujiannya dengan dua objective yang berbeda (*agresif* dan *supportif*) menggunakan NSGA-II :

Tabel 4.14 Tabel Dua Fungsi Objective Kirna 100 Populasi 100 Generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)
96	50.32853	6090	4642.629	162.0165	800.4324	298.7	190.3502	-393.637
97	50.00212	5858.572	156.2828	263.0797	231.7483	166.9176	-1185.19	-38.7265
98	50.00992	5852.538	1624.681	265.5866	248.7508	152.7495	-871.645	-85.3441
99	51.51996	6086.089	2296.702	107.3327	480.5925	293.6552	-572.221	-217.962
100	50.02074	5848.13	1681.773	265.0176	232.4345	151.3324	-874.456	-82.2268
Jumlah							30	70

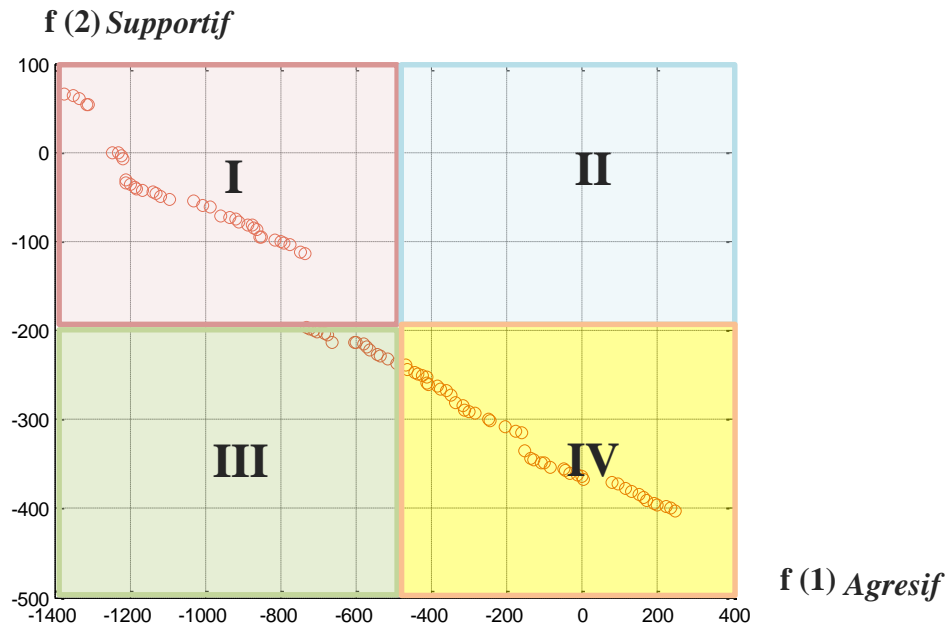
Tabel diatas akan ditunjukkan secara lengkap pada Lampiran 14. Dari beberapa yang diambil dari Tabel 4.14 pada Lampiran 14 didapatkan bahwa seluruh datanya menunjukkan antara dua objective yang berbeda yang merupakan tujuan dari penelitian ini, yaitu antara pahlawan yang tidak hanya dapat menyerang menghancurkan musuhnya namun juga dapat melindungi pasukannya sehingga pasukannya tersebut masih tetap hidup (*supportif*) meskipun ia sendiri harus mati atau mengorbankan nyawanya. Dari keseluruhan data, 100 populasi dengan 20 generasi, didapatkan seluruh populasi menunjukkan Kirna yang sangat *supportif*, karena didapatkan f(2) atau perilaku *supportif* lah yang paling banyak muncul (berjumlah 70 dari 100 populasi).

Seluruh data tersebut akan membentuk sebuah *pareto front* antara dua objective yang berbeda atau saling bertentangan tersebut. Dimana grafik *pareto front* disebut pula *pareto optimal* yaitu kumpulan solusi yang paling optimal dari setiap kondisi yang ditemui oleh Kirna. Grafik *pareto optimal* ditunjukkan oleh Gambar 4.27.



Gambar 4.27 Pareto Optimal Kirna dengan 100 Populasi 100 Generasi

Dari grafik diatas didapat bahwa terjadi penyebaran titik pada *pareto front*, dimana titik – titik tersebut adalah sekumpulan solusi dari setiap kondisi yang ada pada 100 populasi. Populasi dapat dikatakan sebagai individu atau dalam *game* dapat kita sebut sebagai skenario. Jadi dari 100 populasi atau 100 skenario pengujian dengan 100 generasi, 100 generasi adalah 100 kali pembangkitan ulang generasi yang lebih baik, sehingga didapatkan kesimpulan dari grafik tersebut bahwa dengan 100 kali pembangkitan ulang generasi yang lebih baik, terjadi penyebaran titik yang juga lebih baik, hampir memenuhi kondisi *pareto front* atau *pareto optimal* dari dua objective yang saling bertentangan. Karena pada generasi 100 titik tersebar lebih baik dari 75 generasi yaitu titik solusi optimal tersebar merata membentuk garis fungsi minimum NSGA-II yang merupakan *pareto optimal*. Penyebaran tersebut agar lebih mudah membacanya, maka akan dibagi menjadi 4 kuadran pula. berikut tampilan grafik yang telah terbagi menjadi 4 kuadran, ditunjukkan oleh Gambar 4.18 :

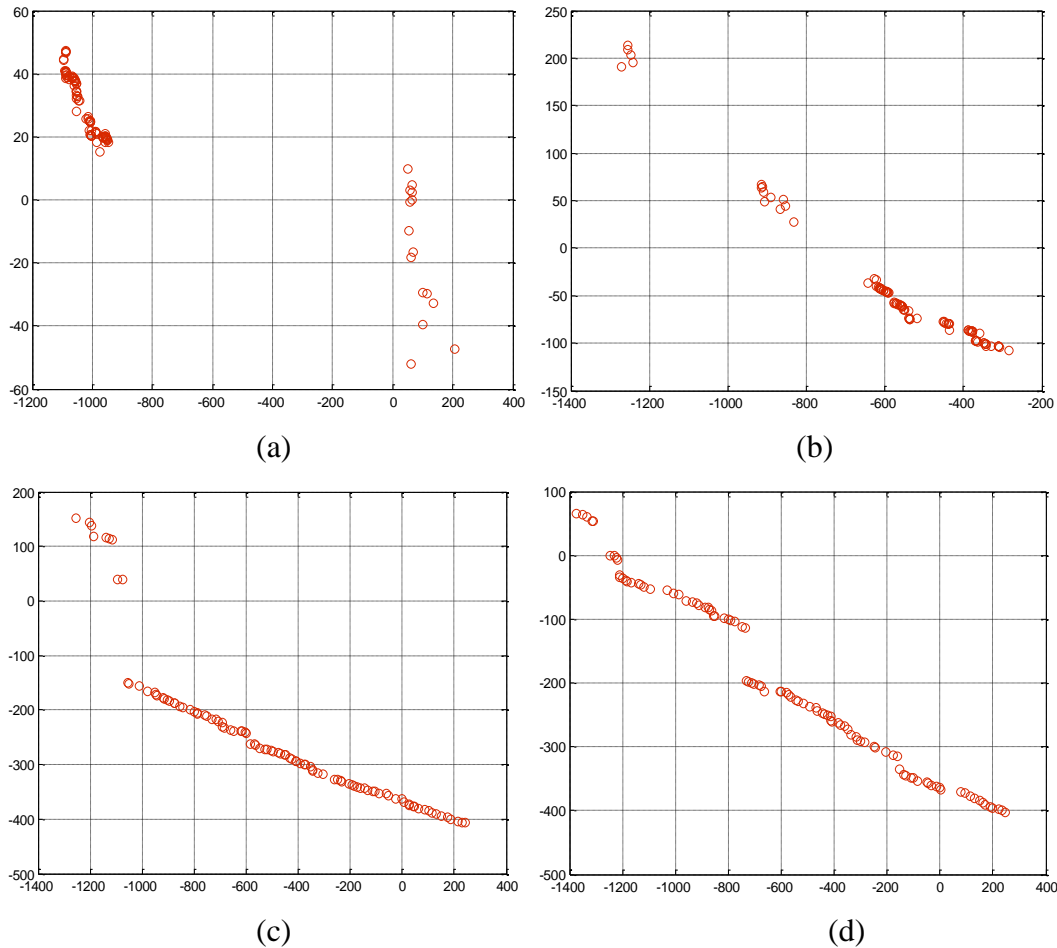


Gambar 4.28 Pembagian 4 kuadran *pareto front* 100 populasi 100 generasi

Maka didapatkan grafik seperti Gambar 4.8 diatas. Grafik tersebut dibagi menjadi 4 kuadran. Kuadran I berwarna merah dimana titik – titik solusi optimal tersebar, menunjukkan perilaku *supportif* yang sangat tinggi namun perilaku *agresif* yang sangat rendah. Titik – titik solusi optimal tersebut sedikit tersebar pada kuadran III yang berwarna hijau, menunjukkan adanya perilaku *supportif* yang sedikit tinggi dan perilaku *agresif* yang sedikit tinggi pula untuk pahlawan. Sedangkan pada kuadran IV yang berwarna kuning inilah titik – titik solusi optimal tersebar lagi yaitu pada perilaku *supportif* yang sangat rendah namun perilaku *agresif* yang sangat tinggi. Dari *pareto optimal* dengan 100 populasi 100 generasi, didapatkan titik – titik solusi optimal hampir rata tersebar pada kuadran I, III, dan IV membentuk garis lurus yang menunjukkan bahwa titik – titik solusi optimal yang juga hampir membentuk *pareto front* fungsi minimum. Sehingga solusi perilaku optimal untuk pahlawan Kirna tersebar diantara perilaku yang sangat tinggi *supportif* nya dan sangat rendah *agresif*nya, yang sedikit tinggi *supportif* dan *agresif*nya, serta yang sangat tinggi *agresif*nya namun sangat rendah *supportif*nya.

Dari keempat pengujian 100 populasi yang berbeda generasi, yaitu 20, 50, 75, dan 100 didapat bahwa sedikitnya generasi (yaitu 20 generasi) belum menunjukkan *pareto optimal* untuk dua objective yang berbeda. Penyebarannya

belum begitu bagus, sehingga dengan percobaan menambah generasi, pada 100 generasi lah titik solusi optimal yang membentuk *pareto front* tersebar hampir sempurna, atau hampir memenuhi kondisi *pareto optimal*. Berikut perbandingan setiap generasi yang diujikan ditunjukkan oleh Gambar 4.29 :



Gambar 4.29 Perbandingan setiap generasi pada 100 populasi

(a) 20 Generasi (b) 50 Generasi (c) 75 Generasi (d) 100 Generasi

Berdasarkan keempat pengujian untuk dua fungsi objective yang berbeda, dimana dua fungsi objective ini merupakan dua tujuan atau dua perilaku pahlawan yang saling bertentangan, yaitu pahlawan yang diinginkan adalah yang tidak hanya dapat menghancurkan musuhnya dan berhasil menang, namun juga pahlawan yang dapat melindungi pasukannya. Data pengujian pada 100 generasilah yang

paling stabil dan dapat membentuk *pareto optimal* dari dua fungsi objective yang saling bertentangan. Sehingga bila terdapat lima perilaku atau lima fungsi objective yang bertentangan pula, maka pengujian lima perilaku ini menggunakan 100 populasi dengan 100 generasi.

Tabel 4.15 Hasil NSGA-II untuk Lima Perilaku Kirna dengan 100 populasi dan 100 generasi

Data ke -	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	f(1)	f(2)	f(3)	f(4)	f(5)
85	50.00028	6090	3355.669	178.4396	431.9578	148.0978	-439.845	-245.362	-516.429	-285.687	-292.682
86	54.05401	5932.669	4800	37.79496	303.22	177.8911	-131.24	-126.479	399.9099	-622.179	-339.777
87	50.02621	6090	30.20532	285.6	14	112.7919	-1523.57	-31.7985	-1925.24	176.3931	456.1546
88	50.89138	5419.582	2505.405	3.4605	825	13.83417	85.46116	-80.0684	-777.658	-786.24	-1390.47
89	53.67716	5483.407	1936.924	3.468158	825	5.325431	1.028631	8.767786	-1000.83	-760.384	-1391.99
90	56.16974	6033.403	4730.394	27.35202	494.3878	166.781	46.70796	-122.011	291.01	-698.228	-653.292
91	50.0652	6090	542.7881	206.552	132.7682	144.8097	-1300.73	-86.5938	-1777.66	-116.207	62.26822
92	53.52638	5483.318	2063.246	3.4605	825	7.307073	23.2814	-0.47319	-943.069	-762.649	-1385.79
93	65.049	5561.114	55.0181	3.4605	627.0633	67.03358	-379.259	416.3576	-1481.8	-559.774	-987.877
94	53.87914	5874.316	4342.084	36.91332	294.1889	163.9115	-210.79	-96.4123	204.7759	-610.895	-354.198
95	50.03138	6090	527.9361	203.4224	158.0414	135.7224	-1278.9	-90.0057	-1799.98	-133.465	8.62481
96	53.52638	5483.318	2063.246	3.4605	825	7.307073	23.2814	-0.47319	-943.069	-762.649	-1385.79
97	50.13285	6090	30	165.0256	177.2787	110.6741	-1356.16	-70.7312	-2087.34	-264.742	-189.406
98	53.97867	5897.105	4457.165	32.36582	287.2324	162.5996	-202.138	-101.598	249.7176	-628.518	-355.481
99	50.72198	5477.958	537.066	225.3058	631.8179	223.0453	-536.285	-32.7993	-1524.03	-15.5952	-426.735
100	50.13047	6086.911	51.36619	278.544	40.72841	122.0091	-1489.28	-36.9953	-1924.34	148.5272	400.4971
Jumlah							3	65	9	1	22

Tabel 4.15 menjelaskan bahwa untuk lima perilaku yang saling bertentangan dapat menjadi semua perilaku optimal dari pahlawan Kirna Waranggani pada 100 populasi atau 100 skenario dengan 100 generasi, didapatkan perilaku optimal yang sering muncul adalah *supportif* yaitu perilaku pahlawan yang melindungi pasukannya agar pasukannya tetap hidup meskipun ia sendiri mati. Pada pengujian lima perilaku dengan 100 populasi 100 generasi ini juga membangkitkan empat perilaku optimal lainnya dari Kirna Waranggani yaitu perilaku *agresif*, *leadership*, *egois 1*, dan *egois 2*. Hanya saja tidak sebanyak perilaku *supportif* saat dihadapkan kondisi atau scenario tersebut. Sebagai contoh, untuk perilaku *agresif* kita ambil salah satu scenario dari pengujian 100 skenario tersebut, dapat dilihat untuk data ke-96, dimana pada saat kondisi kesehatan pahlawan yaitu $x(1)$ berada pada kondisi paling lemah (pahlawan berada pada level 1), dengan melihat kondisi kesehatan musuh atau $x(2)$ yang berada pada mendekati kondisi kesehatan maksimal yaitu bernilai 5483, tidak memungkinkan seorang pahlawan dengan level paling rendah tersebut menyerang sendirian, maka yang perlu dilihat selanjutnya adalah $x(3)$ atau kondisi kesehatan pasukan. Meskipun pada scenario ini pahlawan yang berada pada level yang terendah dengan kesehatan yang sangat rendah pula bila dibandingkan kesehatan musuhnya, dengan memiliki pasukan yang memiliki kesehatan yang besar yaitu bernilai 2063, maka disini keberanian pahlawan muncul, terlebih lagi dengan melihat kondisi selanjutnya yaitu *damage received* pahlawan atau *damage* yang diterima pahlawan $x(4)$ memiliki nilai yang lebih kecil dari *damage* yang akan diterima oleh musuhnya $x(5)$ yaitu hanya 3 sedangkan *damage* yang diterima musuh sangat besar yaitu 825, maka pada kondisi *damage* yang diterima pasukannya juga lebih kecil dari musuhnya pahlawan tersebut pasti akan melakukan atau memilih perilaku *agresif*.

Untuk pengujian scenario yang memunculkan perilaku optimal *supportif* dari pahlawan dapat kita lihat pada skenario ke – 97. Dimana kesehatan pahlawan yang berada pada kondisi minimum kesehatannya, dengan melihat kondisi kesehatan musuhnya $x(2)$ yang sangat besar, pahlawan mulai melihat kondisi kesehatan pasukannya $x(3)$ yang ternyata juga sangat lemah. Dari kondisi tersebut pastinya pahlawan tidak akan menyerang dan tak mungkin melindungi pasukannya, namun dengan melihat kondisi *damage* yang diterima oleh pahlawan lebih kecil

dari musuhnya, maka mulai muncul perilakunya yang ingin melindungi pasukannya meskipun ia sendiri harus mempertaruhkan kehilangan nyawanya. Begitupun dengan pengujian perilaku optimal lain yang muncul pada pahlawan yaitu perilaku *leadership*.

Dapat dilihat pada scenario ke – 98 dimana kesehatannya yang lemah yaitu hanya bernilai 53, ia harus menghadapi musuh dengan kekuatan hampir maksimal yaitu 5897, maka ia akan melihat kondisi pasukannya, pasukannya memiliki nyawa yang cukup besar yaitu 4457. Maka dengan melihat kondisi lain yaitu *damage* yang diterima olehnya dan pasukannya yang lebih rendah dari musuhnya, maka pahlawan akan memilih perilaku *leadership*, yaitu perilaku memimpin, ia akan memimpin pasukannya untuk menyerang bersama – sama dengannya sehingga meskipun ia memiliki nyawa yang lemah, dengan memiliki pasukan yang besar untuk menyerang musuh yang sangat kuat, maka ia akan berperilaku *leadership*.

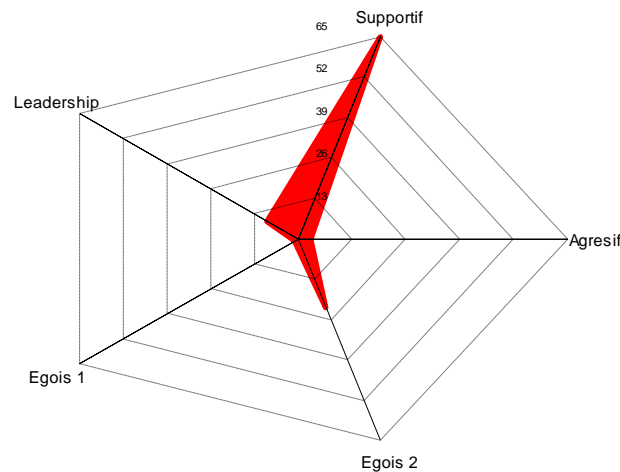
Pahlawan Kirna Warangani memiliki keunikan, dimana dari kelima perilaku yang bertentangan, ia memiliki semua perilaku optimal tersebut saat menemui berbagai macam musuhnya. Tak hanya memiliki keunggulan sifat yaitu *agresif*, *supportif*, dan *leadership*, Kirna juga memiliki keterbatasan yaitu egois yang terdiri dari egois 1 dan egois 2, kedua jenis egois ini telah dijelaskan pada Bab 3. Untuk kondisi yang membuat Kirna berperilaku egois 1, dapat dilihat pada skenario ke-99 dimana kesehatan Kirna yang berada pada titik minimum harus menemui musuh yang memiliki kesehatan yang sangat besar yaitu bernilai 5477, hanya dengan melihat kesehatan dan *damage* yang diterimanya tidak memungkinkan ia untuk menyerang, memimpin, maupun melindungi pasukannya, maka ia memilih egois 1 yang mementingkan dirinya sendiri yaitu melarikan diri. Ia hanya ingin melindungi kesehatannya agar tidak mati. Begitupun dengan keterbatasan Kirna yang lain yaitu perilaku Egois 2 nya yang mementingkan dirinya sendiri dan pasukannya, dimana dalam kondisi ini ia merasa bahwa ia dan pasukannya tidak akan sanggup melawan musuhnya, maka ia lebih memilih untuk melarikan diri dengan mengajak pasukannya melarikan diri bersamanya.

Kondisi tersebut dapat dilihat pada scenario ke-100, dimana kesehatan pahlawan yang berada pada kondisi minimum melawan musuhnya yang memiliki kesehatan yang jauh lebih besar darinya, dan melihat kesehatan pasukannya yang

juga berada pada titik minimal, ia akan mempertimbangkan *damage* yang akan diterimanya dan pasukannya. Dan ternyata *damage* yang akan mereka terima juga lebih besar dari musuhnya, maka pahlawan akan memilih perilaku egois 2 yaitu melarikan diri bersama pasukannya.

Dari keseluruhan hasil pengujian scenario dan mendapatkan beberapa perilaku optimal yang sering muncul, maka dapat ditarik kesimpulan bahwa pahlawan Kirna adalah pahlawan yang sangat *supportif* karena perilaku optimal yang sering muncul padanya saat kondisi tersebut adalah *supportif* meskipun muncul perilaku – perilaku optimal lainnya pada kondisi tertentu.

Berikut radar plot yang menunjukkan bahwa Kirna adalah pahlawan yang sangat *supportif*.



Gambar 4.30 Grafik radar plot kelima perilaku Kirna Warangani

4.2 Simulasi Pada Permainan

Pada permainan DWIPA YUDHA pahlawan memiliki banyak perilaku, dan NSGA-II membantu pahlawan dalam memilih perilaku yang optimal sesuai kondisi pertempuran yang ada. Pada skenario penyerangan pahlawan yang lemah dengan kondisi kesehatannya 50 namun memiliki daya serang yang besar dilihat dari *damage* yang diterimanya hanya 4, memiliki pasukan yang kuat yaitu berjumlah 5 dengan total kondisi kesehatan mereka 2500 dan memiliki daya serang yang juga besar yaitu hanya mendapat *damage* 4 dari musuhnya, ketika menghadapi musuh yang kuat namun daya serangnya yang kecil ditandai dengan *damage* yang diterimanya besar yaitu 825, dengan total kondisi kesehatannya 5400. Maka

perilaku optimal yang dilakukan pahlawan adalah *agresif* yaitu langsung menyerang bangunan utama musuh yaitu townhall. Gambar di bawah ini akan menjelaskan hasil dari perilaku yang dioptimasi dan yang tanpa dioptimasi.



Gambar 4.31 Tampilan Simulasi awal dan akhir perilaku *agresif* pahlawan

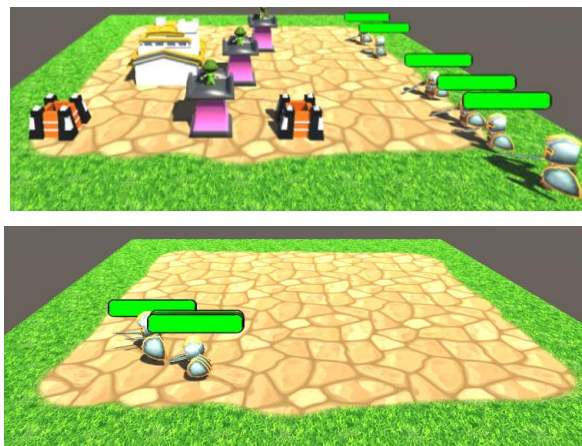
Dari gambar diatas didapatkan tabel data hasil pengujian dengan perilaku *agresif* dilihat dari keberhasilannya dalam penyerangan.

Tabel 4.16 Tabel hasil penyerangan dengan perilaku optimal *agresif*

Health Hero	Health Musuh	Health Pasukan	Damage Received Hero	Damage Received Musuh	Damage Received Pasukan	Tingkat Keberhasilan			
						Sisa Health Hero	Sisa Health Pasukan	Harta yang Didapat	Keterangan
50	5400	2500	4	825	4	50	1882	200	Menghancurkan Townhall pada detik ke 19

Dari data tabel diatas didapatkan, dengan waktu 1 menit penyerangan, pahlawan yang memiliki kesehatan lemah yaitu dengan kondisi kesehatan 50 dengan memiliki pasukan yang kuat yaitu dengan jumlah pasukan 5 dan kondisi total kesehatan mereka 2500, ketika menghadapi musuh yang kuat yaitu dengan kondisi kesehatan 5400, maka tingkat keberhasilan dari perilaku optimal pahlawan dari kondisi penyerangan tersebut yaitu pahlawan yang tetap bertahan hidup dengan kondisi kesehatan utuh, pasukan yang tersisa tetap berjumlah 5 pasukan dengan kesehatan yang berkurang sehingga total kesehatan pasukan 1882. Harta yang

didapatkan adalah 200 dari 200 harta yang dimiliki musuh. Dan dengan perilaku *agresif* ini, pahlawan dapat menghancurkan bangunan utama musuh yaitu townhall pada detik ke 19. Dibandingkan dengan penyerangan tanpa adanya optimasi perilaku didalamnya, maka didapatkan skenario yang sama, namun perilaku yang berbeda. Bila dengan optimasi, perilaku *agresif* membuat pahlawan menyerang townhall sebagai target utamanya, maka dengan penyerangan tanpa perilaku yang dioptimasi, pahlawan menyerang bangunan musuh yang terdekat dengan posisi pertama ketika ia di *deploy*.



Gambar 4.32 Perilaku pahlawan tanpa optimasi dengan kondisi yang sama dengan perilaku optimasi *agresif*

Didapatkan tabel data hasil pengujian perilaku pahlawan tanpa optimasi dengan kondisi yang sama dengan perilaku *agresif* dilihat dari keberhasilannya dalam penyerangan.

Tabel 4.17 Tabel hasil penyerangan biasa dengan kondisi yang sama dengan *agresif*

Health Hero	Health Musuh	Health Pasukan	Damage Received Hero	Damage Received Musuh	Damage Received Pasukan	Tingkat Keberhasilan			
						Sisa Health Hero	Sisa Health Pasukan	Harta yang Didapat	Keterangan
50	5400	2500	4	825	4	50	2146	200	Menghancurkan Townhall pada detik ke 25

Dari tabel diatas, bila dibandingkan dengan tabel perolehan hasil dari perilaku optimal pahlawan yaitu *agresif*, perilaku pahlawan dalam penyerangan yang tanpa dioptimasi memberikan hasil sisa *health* pasukan yang memang lebih banyak daripada perilaku pahlawan yang telah dioptimasi, namun waktu menghancurkan bangunan utama musuh dimana bangunan tersebut merupakan syarat utama pahlawan dapat memenangkan pertempuran ini bila pada perilaku yang tanpa optimasi bangunan utama musuh tersebut hancur di detik ke-25, maka perilaku yang dioptimasi membuat dapat menghancurkan bangunan utama musuh lebih cepat yaitu di detik ke-19.

Pada skenario penyerangan yang lain, pahlawan yang lemah dengan kondisi kesehatannya 53 namun memiliki daya serang yang besar ditandai dengan *damage* yang diterimanya hanya 3,4 memiliki pasukan yang juga lemah dan daya serangnya juga besar yaitu berjumlah 3 dengan total kondisi kesehatan mereka 1900 dengan *damage* yang diterimanya 3,4, menghadapi musuh yang kuat namun daya serangnya kecil yaitu mendapatkan *damage* dari pahlawan dan pasukan sebesar 825 dengan total kondisi kesehatannya 5413. Maka perilaku optimal yang dilakukan pahlawan adalah *supportif* yaitu melindungi pasukannya dengan menambah kesehatan pasukannya tersebut. Gambar di bawah ini akan menjelaskan hasil dari perilaku yang dioptimasi dan yang tanpa dioptimasi.



Gambar 4.33 Tampilan Simulasi awal dan akhir perilaku *supportif* pahlawan

Dari gambar diatas didapatkan tabel data hasil pengujian dengan perilaku *supportif* dilihat dari keberhasilannya dalam penyerangan.

Tabel 4.18 Tabel hasil penyerangan dengan perilaku optimal *supportif*

Health Hero	Health Musuh	Health Pasukan	Damage Received Hero	Damage Received Musuh	Damage Received Pasukan	Tingkat Keberhasilan			
						Sisa Health Hero	Sisa Health Pasukan	Harta yang Didapat	Keterangan
53	5400	1900	3,4	825	3,4	53	1822	100	Dari 3 pasukan yang dimiliki, diakhir penyerangan tetap 3 pasukan

Dari data tabel diatas didapatkan, dengan waktu 1 menit penyerangan, pahlawan yang memiliki kondisi tersebut, dengan memilih perilaku yang optimal yaitu *supportif*, maka tingkat keberhasilan dari perilaku optimal pahlawan dari kondisi penyerangan tersebut yaitu pahlawan yang tetap bertahan hidup dengan kondisi kesehatan utuh dengan dapat melindungi pasukannya dilihat dari pasukan yang tersisa setelah penyerangan tetap berjumlah 3 pasukan dengan kesehatan yang berkurang hanya 25 dari total keseluruhan kesehatan awal pasukan, sehingga total kesehatan pasukan di akhir penyerangan menjadi 1875. Harta yang didapatkan adalah 100 dari 200 harta yang dimiliki musuh, menandakan dengan pahlawan yang fokus dalam melindungi pasukannya, melalaikan atau tidak mengejar keseluruhan harta musuh, hanya fokus melindungi pasukannya agar semuanya tetap bertahan hidup atau darah yang berkurang tidak terlalu banyak dibandingkan dengan penyerangan tanpa adanya optimasi perilaku didalamnya. Pada hal ini didapatkan skenario yang sama, namun perilaku yang berbeda. Bila dengan optimasi, perilaku *supportif* membuat pahlawan dapat melindungi pasukannya, maka dengan penyerangan tanpa perilaku yang dioptimasi, pahlawan hanya fokus menyerang menghancurkan bangunan musuh dan mengambil harta musuhnya sebanyak – banyaknya tanpa menghiraukan pasukannya.



Gambar 4.34 Perilaku pahlawan tanpa optimasi dengan kondisi yang sama dengan perilaku optimasi *supportif*

Didapatkan tabel data hasil pengujian perilaku pahlawan tanpa optimasi dengan kondisi yang sama dengan perilaku *supportif* dilihat dari keberhasilannya dalam penyerangan.

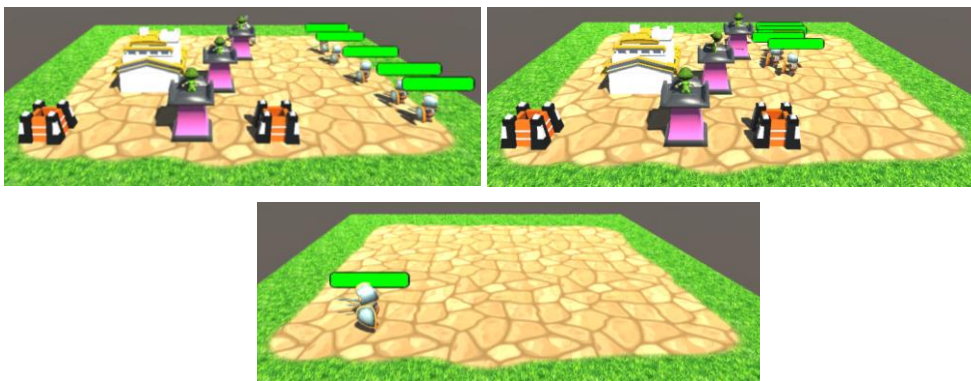
Tabel 4.19 Tabel hasil penyerangan biasa dengan kondisi yang sama dengan *supportif*

Health Hero	Health Musuh	Health Pasukan	Damage Received Hero	Damage Received Musuh	Damage Received Pasukan	Tingkat Keberhasilan			
						Sisa Health Hero	Sisa Health Pasukan	Harta yang Didapat	Keterangan
53	5400	1900	3,4	825	3,4	53	1576	200	Dari 3 pasukan yang dimiliki, diakhir penyerangan tetap 3 pasukan

Dari tabel diatas, bila dibandingkan dengan tabel perolehan hasil dari perilaku optimal pahlawan yaitu *supportif*, perilaku pahlawan dalam penyerangan yang tanpa dioptimasi memberikan hasil harta yang didapatkan oleh penyerangan biasa memang lebih banyak daripada perilaku pahlawan yang telah dioptimasi,

namun sisa kesehatan pasukan saat akhir penyerangan berkurang lebih banyak daripada sisa kesehatan yang berkurang pada perilaku pahlawan yang dioptimasi. Bila pada perilaku pahlawan yang dioptimasi, kesehatan seluruh pasukan hanya berkurang 25 dari total keseluruhan kondisi awal kesehatan pasukan sebelum penyerang, dengan perilaku pahlawan tanpa optimasi kesehatan seluruh pasukan berkurang lebih banyak yaitu 326.

Pada skenario penyerangan yang lain, pahlawan yang lemah dengan kondisi kesehatannya 54 dan memiliki daya serangan yang kuat dilihat dari *damage* yang diterimanya yaitu 37 dari *damage* yang diterima musuhnya yaitu 303, memiliki pasukan yang kuat dengan daya serangan yang kuat pula, dimana pasukan tersebut berjumlah 5 dengan total kondisi kesehatan mereka maksimum yaitu 4800, menghadapi musuh yang kuat dengan total kondisi kesehatannya 5200 namun daya serangnya yang kecil dilihat dari *damage* yang diterimanya lebih besar dari yang diterima pahlawan ataupun pasukan. Maka perilaku optimal yang dilakukan pahlawan adalah *leadership* yaitu memimpin pasukannya untuk menyerang secara bersama – sama bangunan musuh yang menjadi targetnya. Tampak jelas pada gambar dibawah ini, dimanapun pasukan yang diletakkan oleh pemain atau yang sesuai dengan keinginan pemain, pasukan tersebut tetap akan bergerak menuju kearah pahlawan dimana pasukan tersebut akan langsung menyerang bangunan musuh yang diserang oleh pahlawan. Dengan kata lain, pasukan akan menyerang target yang sama dengan target pahlawan, ini demi melindungi pahlawan yang bernyawa lemah agar tidak cepat mati.



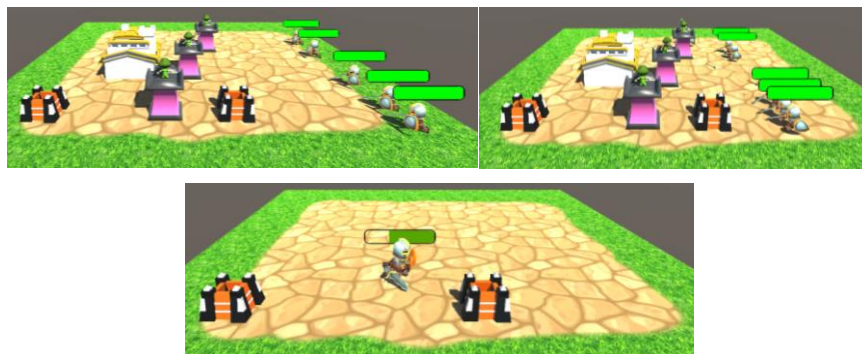
Gambar 4.35 Tampilan Simulasi awal dan akhir perilaku *leadership* pahlawan

Dari gambar diatas didapatkan tabel data hasil pengujian dengan perilaku *leadership* dilihat dari keberhasilannya dalam penyerangan.

Tabel 4.20 Tabel hasil penyerangan dengan perilaku optimal *leadership*

Health Hero	Health Musuh	Health Pasukan	Damage Received Hero	Damage Received Musuh	Damage Received Pasukan	Tingkat Keberhasilan			
						Sisa Health Hero	Sisa Health Pasukan	Harta yang didapat	Keterangan
54	5900	4800	37	303	37	54	1180	200	Pahlawan bertahan hidup

Dari data tabel diatas didapatkan, dengan waktu 1 menit penyerangan, pahlawan yang memiliki kondisi tersebut, dengan memilih perilaku yang optimal yaitu *leadership*, maka tingkat keberhasilan dari perilaku optimal pahlawan dari kondisi penyerangan tersebut yaitu pahlawan yang tetap bertahan hidup dengan kondisi kesehatan utuh dengan dapat mengambil keseluruhan harta milik musuhnya, namun pahlawan yang berperilaku *leadership* ini tidak dapat melindungi pasukannya, dilihat dari pasukan yang tersisa setelah penyerangan berjumlah 2 dari 5 pasukan awal pertempuran. Pada hal ini didapatkan skenario yang sama, namun perilaku yang berbeda. Bila dengan optimasi, perilaku *leadership* membuat pahlawan dapat bertahan hidup dan mengambil keseluruhan harta milik musuhnya, maka dengan penyerangan tanpa perilaku yang dioptimasi, pahlawan dan pasukan fokus menyerang menghancurkan bangunan musuh yang berada terdekat dari posisi ia di-*deploy* pemain, dengan perilaku yang menyerang secara individual ini, pahlawan tidak menghiraukan bila kesehatannya lemah meskipun daya serangnya sedikit kuat.



Gambar 4.35 Tampilan Simulasi awal dan akhir perilaku *leadership* pahlawan

Dari gambar diatas didapatkan tabel data hasil pengujian dengan perilaku *leadership* dilihat dari keberhasilannya dalam penyerangan.

Tabel 4.21 Tabel hasil penyerangan biasa dengan kondisi yang sama dengan *leadership*

Health Hero	Health Musuh	Health Pasukan	Damage Received Hero	Damage Received Musuh	Damage Received Pasukan	Tingkat Keberhasilan			
						Sisa Health Hero	Sisa Health Pasukan	Harta yang Didapat	Keterangan
54	5900	4800	37	303	37	0	72	63	Pahlawan mati

Dari tabel diatas, bila dibandingkan dengan tabel perolehan hasil dari perilaku optimal pahlawan yaitu *leadership*, perilaku pahlawan dalam penyerangan yang tanpa dioptimasi memberikan hasil pahlawan yang mati atau tidak dapat bertahan hidup, dimana perilaku pahlawan yang tanpa dioptimasi tersebut juga membuat banyak pasukan yang mati, dilihat dari sisa kesehatan pasukan di akhir pertempuran yaitu hanya tersisa 72 dari total kesehatan pasukan 4800. Tak hanya itu, harta yang didapatkan oleh penyerangan biasa dengan kondisi tersebut membuat pahlawan maupun pahlawan tidak dapat mengambil banyak harta musuhnya, hanya 63 dari total harta musuh yaitu 200.

Pada skenario penyerangan lain, pahlawan yang sangat lemah dengan kondisi kesehatannya 50 memiliki daya serang yang kecil dilihat dari *damage* yang diterimanya yaitu 225, dengan tanpa melihat keadaan pasukannya ketika menghadapi musuh yang kuat dengan total kondisi kesehatan 5400 dengan daya serang yang kecil dilihat dari *damage* yang diterimanya lebih besar dari yang diterima pahlawan ataupun pasukan yaitu 631. Maka perilaku optimal yang dilakukan pahlawan adalah melarikan diri atau egois 1, yaitu ketika di-*deploy* oleh pemain ia akan langsung lari menuju tempat yang aman. Melarikan diri atau egois 1 ini bertujuan menjaga keutuhan kondisi kesehatan pahlawan, karena melihat kondisi yang sangat tidak memungkinkan untuk menyerang, maka pahlawan akan memilih melarikan diri. Perilaku ini tidak mendapatkan keuntungan apapun, malah

membunuh semua pasukan, dan harta yang didapat hanya 4 dari 200 harta yang dimiliki musuh.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

NSGA-II dapat digunakan untuk memilih perilaku pahlawan secara otomatis. Metode ini berhasil memilih perilaku pahlawan yang sesuai dengan kondisi pertempuran yang dihadapi. Simulasi yang dilakukan pada skenario yang berbeda dapat membuktikan bahwa NSGA-II memberikan hasil yang optimal di setiap skenario tersebut.

Pahlawan dalam *game* ini memiliki kondisi kesehatan yang lemah, bila tidak ditambahkan dengan kondisi kesehatan pasukannya dan tanpa melihat *damage* yang diterimanya, pahlawan pasti akan kalah ketika menemui musuh yang kuat. Berdasarkan hal tersebut, agar tidak kalah ketika menyerang, kondisi – kondisi yang ada dijadikan parameter dan akan dibandingkan antara satu dengan yang lainnya, kemudian dijadikan masukan pada NSGA-II karena NSGA-II mampu mengoptimalkan pemilihan perilaku pahlawan dengan melihat beberapa parameter tersebut. Pada penelitian ini 6 parameter yang digunakan adalah kesehatan pahlawan, kesehatan musuh, kesehatan pasukan, *damage* yang diterima pahlawan, *damage* yang diterima musuh, dan *damage* yang diterima oleh pasukan. Parameter ini memiliki besar nilai yang berbeda pada setiap penyerangan, oleh karena itu dibentuk beberapa fungsi objective. Setiap fungsi objective memiliki persamaan yang merepresentasikan satu tujuan utama penyerangan yang hendak dicapai. Fungsi objective NSGA-II ini akan menghitung nilai-nilai parameter tersebut menggunakan persamaan yang telah dibentuk, kemudian memilih nilai fungsi objective maksimal pada kondisi yang dihadapi pahlawan pada saat itu sebagai dasar pemilihan perilaku yang akan digunakan untuk menyerang.

Pada skenario penyerangan pahlawan yang memiliki kesehatan lemah namun daya serangnya besar, memiliki pasukan dengan kesehatan yang kuat dan daya serangnya besar ketika menghadapi musuh yang memiliki kesehatan kuat namun daya serangnya kecil, diujikan dengan menggunakan NSGA-II perilaku optimal yang dilakukan pahlawan adalah *agresif* yaitu langsung menyerang

bangunan utama musuh yaitu townhall. Perilaku ini menunjukkan bahwa pahlawan dapat menghancurkan bangunan utama musuh tersebut 42% lebih cepat daripada penyerangan yang dilakukan tanpa optimasi pemilihan perilaku.

Pada skenario penyerangan yang lain, pahlawan yang memiliki kesehatan lemah namun daya serangnya besar, memiliki pasukan dengan kesehatan yang juga lemah namun daya serangnya besar ketika menghadapi musuh yang memiliki kesehatan kuat namun daya serangnya kecil, diujikan dengan menggunakan NSGA-II perilaku optimal yang dilakukan pahlawan adalah *supportif* yaitu melindungi pasukannya dengan menambah kesehatan pasukannya tersebut. Dengan perilaku tersebut kesehatan pasukan hanya berkurang 5% dari total kesehatan yang dimiliki, sedangkan penyerangan yang dilakukan tanpa optimasi kesehatan pasukan berkurang 18% dari total kesehatan yang dimiliki.

Pada skenario penyerangan yang lain, pahlawan yang memiliki kesehatan lemah dan daya serangnya menengah, memiliki pasukan dengan kesehatan yang sangat kuat dan daya serangnya besar pula ketika menghadapi musuh yang kuat namun daya serangnya kecil, maka perilaku optimal yang dilakukan pahlawan adalah *leadership* yaitu memimpin pasukannya untuk menyerang secara bersama – sama bangunan musuh yang menjadi targetnya. Dengan perilaku tersebut, pahlawan dapat bertahan hidup hingga akhir penyerangan dan harta yang didapatkan lebih banyak 68,6% daripada penyerangan yang dilakukan tanpa optimasi.

Keseluruhan perilaku optimal yang dilakukan pahlawan, memiliki tujuan memaksimalkan keuntungan yang diperoleh pada saat penyerangan. Perilaku *agresif* dapat mempercepat penghancurkan bangunan utama musuh sehingga kemenangan dapat diraih hanya dalam waktu beberapa detik, perilaku *supportif* dapat melindungi pasukan agar tetap hidup sehingga dengan pasukan yang ada, dapat membantu pahlawan menghancurkan seluruh bangunan musuh. Pada perilaku *leadership*, meskipun pahlawan dalam keadaan lemah dan tidak mampu melawan musuh, dengan perilaku *leadership* mampu membuat pasukan menyerang secara bersama – sama sehingga pahlawan dapat bertahan hidup hingga akhir pertempuran.

5.2 Saran

Optimasi perilaku pahlawan dalam permainan DWIPA YUDHA ini merupakan salah satu topik riset dalam permainan strategi. Beberapa saran untuk pengembangan sistem terutama berhubungan dengan optimasi perilaku pahlawan dan NSGA-II, yaitu:

1. Membuat pahlawan dalam *game* lebih unik karena memiliki kepribadian dan berperilaku layaknya manusia atau pahlawan dalam dunia nyata maupun dalam film.
2. Membuat pahlawan dapat memenangkan pertempuran dengan tetap mempertahankan nyawanya dan nyawa pasukannya.
3. Melakukan percobaan dengan model multiobyektif lainnya sehingga bisa membandingkan performa *pareto optimal* dan grafik radar terhadap berbagai permasalahan yang dihadapi.
4. Menambah perilaku pahlawan agar muncul lebih banyak variasi perilaku optimal pahlawan pada saat operasi penyerangan.

DAFTAR PUSTAKA

- [1] Camerer, Colin, Teck Ho, and Kuan Chong. "Models of thinking, learning, and teaching in games." *American Economic Review* (2003): 192-195.
- [2] J. B. Ahlquist and J. Novak, Game Artificial Intelligent, ser. Game Development Essentials. Canada: Thompson Learning, 2008.
- [3] Fernández-Ares, Antonio, et al. "Optimizing player behavior in a real-time strategy game using evolutionary algorithms." *Evolutionary Computation (CEC), 2011 IEEE Congress on. IEEE*, 2011.
- [4] Rinanto, Noorman. "Optimasi Strategi Menyerang NPC Group Pada Game Pertarungan Jarak Dekat Menggunakan Simulated Annealing." *Optimasi Strategi Menyerang NPC Group Pada Game Pertarungan Jarak Dekat Menggunakan Simulated Annealing* (2012).
- [5] Miles, Chris, and Sushil J. Louis. "Case-Injection Improves Response Time for a Real-Time Strategy Game." *CIG*. 2005.
- [6] Purba, Kristo Radion, Rini Nur Hasanah, and M. Aziz Muslim. "Implementasi Logika Fuzzy Untuk Mengatur Perilaku Musuh dalam Game Bertipe Action-RPG." *Jurnal EECCIS 7.1* (2013): pp-15.
- [7] Haryono, Alm, and Ari Utomo. "Optimalisasi Penerimaan Pajak Hiburan Dalam Rangka Meningkatkan Pendapatan Asli Daerah (Studi Pada Pemerintah Kota Bandar Lampung Tahun 2011)." (2013).
- [8] Muhammad, Aidi. "Analisis Optimalisasi Pelayanan Konsumen Berdasarkan Teori Antrian Pada Kaltimgps. Com Di Samarinda." (2014).
- [9] Yukl, Gary. "An evaluative essay on current conceptions of effective leadership." *European Journal of Work and Organizational Psychology* 8.1 (1999): 33-48.
- [10] Laksma Parameshta, Dominico. *Penerapan Multi-Objective Particle Swarm Optimization Untuk Kasus Capacitated Vehicle Routing Problem Dengan Load Balancing*. Diss. Uajy, 2014.
- [11] Deb, K., PRATAP, A., AGARWAL, S., AND MEYARIVAN, T. A fast and elitis multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on* 6, 2 (2002), 182-197.

- [12] Rifki, Omar, and Hirotaka Ono. "Robustness Analysis of Evolutionary Algorithms to Portfolio Optimization Against Errors in Asset Means." *Operations Research Proceedings 2013*. Springer International Publishing, 2014. 369-375.
- [13] Widiastuti, Ika. "Pemilihan Perilaku NPC Pada Game Pertarungan Jarak Dekat Menggunakan Fuzzy Coordinator." *Pemilihan Perilaku NPC Pada Game Pertarungan Jarak Dekat Menggunakan Fuzzy Coordinator* (2012).
- [14] Supeno Mardi S.N, Yunifa Miftachul Arif, Mochamad Hariadi, Mauridhi H.P (2011). *Perilaku Taktis untuk Non Player Character di Game Peperangan Meniru Strategi Manusia menggunakan Fuzzy Logic dan Hierarchical Finite State Machine*. Jurnal Ilmiah Kursor vol 6. No 1.
- [15] Sarinastiti, Widi. "Animasi Perilaku Pasukan Pada Game Real Time Strategy Menggunakan Flocking Behaviour." *Animasi Perilaku Pasukan Pada Game Real Time Strategy Menggunakan Flocking Behaviour* (2014).
- [16] Prajitno, Mohamad Iman. "Simulasi Pengambilan Keputusan Pada Kasus Multiobjective Menggunakan Agen Cerdas Dan Hierarchical Finite State Machine." *Simulasi Pengambilan Keputusan Pada Kasus Multiobjective Menggunakan Agen Cerdas Dan Hierarchical Finite State Machine* (2009).
- [17] Arif, Yunifa Miftachul. "Strategi Menyerang Pada Game FPS Menggunakan Hierarchy Finite State Machine Dan Logika Fuzzy." *Strategi Menyerang Pada Game FPS Menggunakan Hierarchy Finite State Machine Dan Logika Fuzzy* (2010).
- [18] Fernando, Ray. *Perancangan Program Aplikasi Optimasi Listrik pada Industri Plastik Menggunakan Metode Sequential Dynamic Programming*. Diss. BINUS, 2011.
- [19] Arvanitis, K. G., P. N. Paraskevopoulos, and A. A. Vernardos. "Multirate adaptive temperature control of greenhouses." *Computers and Electronics in Agriculture* 26.3 (2000): 303-320.
- [20] Haji, Son. "Aplikasi Particle Swarm Optimization (PSO) untuk Optimasi Parameter Kontroler Pid pada Sistem: Kendali Iklim Greenhouse." *Prosiding*

Seminar Nasional Teknik Elektro Politeknik Negeri Jakarta (Snte 2014):ISBN 978-979-3288-95-6.

- [21] Jost, Monika. "Behaviour Modification." (2000).
- [22] McCrae, Robert R., and Paul T. Costa. "Validation of the five-factor model of personality across instruments and observers." *Journal of personality and social psychology* 52.1 (1987): 81.
- [23] Buss, David M. "Evolutionary psychology: A new paradigm for psychological science." *Psychological inquiry* 6.1 (1995): 1-30.
- [24] Dulwich College Suzhou-The Characteristic of Passive, Aggressive, and Assertive Communication.
- [25] Geen, Russell G., and Edward D. Donnerstein, eds. *Human aggression: Theories, research, and implications for social policy*. Elsevier, 1998.
- [26] Retno Habsari, Ari. 2008. *Terobosan Kepemimpinan (Panduan Pelatihan Kepemimpinan)*. Yogyakarta:Media Pressindo.
- [27] Rolliawati, Dwi. "Autonomous Agen Pembeli Dan Penjual Dengan Pendekatan Multiobjective Optimization Using Evolutionary Algorithm (MOEA) NSGA-II." *Autonomous Agen Pembeli Dan Penjual Dengan Pendekatan Multiobjective Optimization Using Evolutionary Algorithm (MOEA) NSGA-II* (2011).
- [28] Deb, Kalyanmoy, et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *Evolutionary Computation, IEEE Transactions on* 6.2 (2002): 182-197.
- [29] David Goldberg, "The Design of Innovation (Genetic Algorithms and Evolutionary Computation) ", Springer; 1 edition (June 30, 2002).
- [30] Tobing, Reynold Lumban. "Sistem Simulasi Penjadwalan Kuliah Dengan Menggunakan Algoritma Genetik." (2010).
- [31] Kukuh Adisusilo, Anang. "Optimasi Perilaku Agen Pada Permainan Tinju Menggunakan Algoritma Genetika" *Optimasi Perilaku Agen Pada Permainan Tinju Menggunakan Algoritma Genetika* (2013).
- [32] Srinivas, Nidamarthi, and Kalyanmoy Deb. "Multiobjective optimization using nondominated sorting in genetic algorithms." *Evolutionary computation* 2.3 (1994): 221-248.

- [33] Deb K. (1995), *Optimization for engineering design: algorithms and examples*, Prentice-Hall, New Delhi, India.
- [34] Ariyadi. "Evolusi Penyerangan Creep Untuk Pengaturan Tingkat Kesulitan Dinamis Pada Permainan Tower Defense Dengan Metode NSGA-II"*Evolusi Penyerangan Creep Untuk Pengaturan Tingkat Kesulitan Dinamis Pada Permainan Tower Defense Dengan Metode NSGA-II*(2015).
- [35] Cahyono Dwi, Agen Cerdas untuk Asperger Syndrom, Tesis, Elektro-ITS, 2009.

LAMPIRAN 1

Tabel Hasil 4.1 Tabel Fungsi Objective NSGA-II Batara dengan 100 Populasi 20 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif
105.6348	5720.096	2945.98	34.79147	1236.62	116.5941	1338.141	688.085
105.0556	5686.558	2904.265	45.56407	1160.438	125.1436	1310.743	694.774
100	5373.654	1231.085	135.5717	282.642	51.32905	879.4738	941.9205
105.6653	5727.796	3119.139	42.14022	1180.434	115.0937	1341.68	685.9158
105.5139	5688.936	3237.4	33.44251	1213.547	122.9075	1376.083	673.8364
105.6338	5721.62	2941.878	36.51874	1234.231	116.2135	1336.331	688.6496
105.3129	5722.86	2496.436	57.0474	1097.094	110.2254	1239.428	737.2333
104.9917	5718.456	2642.574	38.71557	1104.269	110.5056	1251.147	718.99
105.3967	5694.743	3321.777	52.96455	1226.518	102.4378	1383.17	671.6599
105.7628	5711.78	3109.9	36.94773	1212.188	118.7537	1358.458	685.1195
105.9629	5765.828	2823.141	29.68532	1225.781	120.4756	1311.168	693.6825
100	5385.465	1223.574	138.361	318.7458	54.73671	882.2985	932.5643
105.3561	5723.008	2503.205	55.26428	1051.71	110.7717	1230.145	744.9691
100.1645	5380.75	980.8329	126.4952	152.2216	54.6415	819.725	980.6707
104.9797	5672.606	3900.744	42.95692	1245.489	113.4665	1452.506	620.217
105.1974	5693.29	3725.822	44.37892	1219.36	113.9546	1423.071	636.5185
105.3675	5723.859	2548.863	56.28368	1014.486	110.4908	1226.116	748.3703
105.3617	5722.934	2360.563	55.30124	1058.323	110.6314	1215.361	753.5791
104.9753	5671.399	4085.253	43.07016	1242.856	113.528	1473.759	608.498
104.972	5674.054	3914.307	43.2844	1243.265	113.6282	1452.658	619.1456
105.3039	5721.243	2497.98	55.37305	1042.791	110.6812	1226.505	746.0562
100	5374.258	1007.818	138.49	191.6451	52.36658	830.283	971.5538
105.495	5707.301	3088.931	35.81042	1247.79	119.6991	1358.825	675.0772
105.4432	5734.228	2580.235	60.80765	1041.401	110.4327	1234.134	741.5005
105.3119	5673.199	3451.327	48.07286	1243.068	108.4869	1409.174	660.259
105.3971	5695.493	3307.296	53.13044	1226.559	102.4541	1381.169	672.4646
100.0139	5377.042	1240.947	133.179	280.7086	48.16322	879.2749	942.3633
104.9616	5668.765	3881.114	45.08557	1246.881	116.9094	1451.502	620.3921
104.9812	5672.344	3857.469	43.02559	1237.614	113.7511	1445.635	624.3895
104.8945	5648.32	3516.829	51.22038	1250	106.2307	1416.779	651.8337
105.6309	5720.026	2939.66	35.11715	1250	116.6337	1340.652	686.2048
105.5123	5689.364	3238.605	33.48118	1211.015	122.8762	1375.357	674.0638
104.9759	5672.618	3896.584	42.84877	1245.713	113.4814	1451.961	620.3575
105.3123	5721.425	2638.315	56.70839	1053.625	109.1164	1245.766	735.7627
104.8246	5724.04	2865.073	43.85149	1081.5	114.217	1263.685	701.6922
104.9932	5717.612	2641.382	39.15386	1103.262	110.7371	1251.144	719.3788

105.0541	5687.402	2905.457	45.12578	1161.445	124.9122	1310.746	694.3852
105.2088	5678.237	4035.735	42.32067	1247.274	113.6058	1473.151	614.7652
100	5373.164	865.4592	126.0065	149.4937	54.86372	803.9434	986.6419
100	5396.089	1081.986	141.3857	223.1754	52.7313	837.2496	956.617
104.9596	5674.261	3882.695	42.53547	1244.891	113.8431	1448.933	620.5323
105.39	5704.798	3352.508	54.33763	1232.147	102.4699	1383.557	666.3984
100	5347.853	1354.563	120.0903	345.6809	42.63735	921.6253	931.6198
105.5141	5688.892	3227.547	33.35055	1212.827	122.907	1374.783	674.6234
100	5375.106	1256.09	139.4437	185.5716	50.92498	857.3784	956.4792
105.2565	5697.315	3357.717	49.60823	1247.5	104.5349	1387.33	661.0133
100.025	5396.638	1116.412	140.2296	203.9341	47.80815	837.1716	959.9666
104.9991	5719.008	2655.763	38.32662	1107.998	110.4492	1253.616	717.5401
104.8308	5724.805	2857.13	44.69798	1081.206	113.5516	1262.533	702.5635
104.9792	5672.397	3854.141	42.89488	1242.547	113.4084	1446.414	623.8698
104.9922	5718.665	2689.177	38.7776	1107.211	110.5638	1257.239	715.3373
100.0065	5385.48	1224.937	138.5922	311.3908	54.7381	880.7983	933.8523
100.1581	5380.735	979.4694	126.264	159.5766	54.64011	821.2252	979.3828
105.5896	5717.856	2944.739	35.03935	1234.268	116.4763	1337.019	688.0893
100.7544	5375.897	466.2999	151.6089	282.2985	46.71769	811.4412	1012.174
104.9733	5672.091	3898.718	42.6159	1244.463	113.3403	1452.062	620.5279
100.1773	5388.186	1215.539	133.4851	334.5225	53.25582	889.657	934.2435
105.3605	5726.263	2577.911	56.25935	1007.82	110.336	1226.592	746.9291
104.835	5724.07	2866.719	43.92118	1081.925	114.1584	1264.283	701.7674
105.1008	5688.798	2905.505	45.31619	1162.79	125.2615	1311.865	694.7697
100.2074	5380.715	980.8041	126.6343	151.9537	54.6654	820.9496	981.6857
100.0999	5392.976	1092.721	140.3131	209.0187	51.40117	839.3895	961.684
100.5692	5385.77	921.392	127.6466	140.98	55.80032	819.8673	994.099
105.5293	5733.107	2731.701	55.23644	1064.378	108.9914	1260.852	730.0756
104.9761	5672.232	3881.902	42.44515	1244.819	113.6094	1450.206	621.4997
105.3014	5671.423	3710.81	48.29219	1249.776	115.2799	1441.345	639.1716
100.0491	5347.656	1253.067	135.4836	224.8004	50.97959	880.3314	956.9742
100	5363.014	1167.768	135.2651	203.9876	56.17613	856.8825	959.5211
105.3687	5671.497	3650.301	50.61024	1247.534	105.1095	1435.974	649.431
105.7531	5711.245	3124.758	36.69863	1211.808	118.8956	1360.041	684.0182
100.1383	5382.777	1217.183	141.5175	303.3604	52.45176	882.9903	940.3579
100.0173	5350.916	1300.548	128.8013	219.5951	52.2541	882.3824	952.3888
105.3517	5721.537	2578.577	55.08172	1063.024	110.6854	1242.282	738.3055
105.5065	5688.384	3224.211	33.83146	1209.818	122.9639	1373.615	675.2863
105.6673	5728.747	3117.372	42.35215	1178.756	114.7323	1340.707	686.3134
105.9761	5718.931	3121.175	33.03788	1232.062	122.7087	1368.04	682.4615
105.6365	5675.141	3369.731	36.50242	1250	115.6803	1410.428	667.8493

105.6469	5744.273	2700.832	61.23516	1053.392	111.0706	1252.923	733.6237
105.2976	5671.435	3706.65	48.18404	1250	115.2947	1440.8	639.3121
105.4039	5694.871	3357.659	53.21393	1234.795	102.6172	1389.57	667.9591
105.3525	5692.431	2850.813	51.27126	1236.636	119.1234	1329.928	693.8389
105.2409	5652.576	3437.325	51.57016	1250	108.0034	1415.979	663.2836
105.3666	5673.379	3597.448	50.76507	1245.799	105.2139	1428.485	652.7514
105.3413	5671.4	3615.993	47.73209	1225.214	103.0051	1425.778	655.5759
100	5373.945	1199.106	137.3288	325.3266	54.79797	886.1136	935.5177
105.7185	5726.938	3002.464	43.14112	1186.555	115.9357	1331.506	693.7572
105.3995	5694.884	3304.961	52.7938	1226.874	102.7069	1381.315	672.6317
100	5363.768	1185.906	134.5441	179.2197	57.51858	852.4591	961.6811
105.6353	5720.009	2931.882	34.66948	1236.755	116.6031	1336.589	689.0237
105.1887	5654.456	3561.738	49.93856	1182.396	101.3004	1411.492	667.3875
105.3573	5721.728	2545.071	55.41448	1055.69	110.6929	1236.614	741.8601
105.3409	5672.288	3559.889	47.66967	1225.618	103.1542	1418.932	658.9807
105.341	5670.484	3573.97	47.58819	1225.194	103.0854	1421.26	658.5313
105.3676	5695.949	3351.696	53.33787	1221.875	102.7361	1384.082	669.4173
105.6501	5744.948	2715.503	61.46671	1054.785	111.1077	1254.779	732.3368
105.4932	5734.274	2580.261	60.81125	1041.426	110.4674	1235.621	742.5943
105.5582	5735.878	2691.074	58.64595	1072.114	112.1975	1257.492	730.3792
105.4366	5727.155	2502.121	53.93346	1055.651	111.1173	1231.655	745.0874
105.4133	5695.134	3321.63	52.29082	1225.155	102.169	1383.171	672.2629
100.0002	5363.769	1187.579	134.5678	179.5871	57.69037	852.7443	961.4415
Jumlah						76	24

Lampiran 2

Tabel Hasil 4.2 Tabel Hasil 2 Fungsi Objective NSGA-II Batara dengan 100 Populasi 50 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif
103.9809	5742.132	439.2638	123.274	473.7485	226.8816	789.1828	898.8292
102.0957	5575.356	3410.448	135.526	419.0709	213.7271	1137.895	709.6743
104.0433	5739.157	373.9048	123.6941	474.0666	227.6596	784.7589	904.8793
103.9768	5738.918	454.7233	122.6394	474.0806	226.5729	792.3698	898.4437
104.1479	5826.18	1241.026	128.0275	504.5799	230.0692	858.7531	824.6958
104.1418	5826.321	1224.05	128.0786	504.8403	230.081	856.5923	825.6146
104.7065	5826.162	771.179	128.4402	491.2658	231.9281	817.2985	870.0559
102.5352	5565.408	4143.982	135.4654	423.1674	213.0406	1242.018	672.4159
102.5748	5561.618	3882.654	135.4692	423.0631	213.0506	1214.334	691.5641
102.191	5576.051	2360.99	135.6665	430.0133	214.2301	1020.73	779.6056
104.0122	5737.893	475.843	123.3381	475.5309	226.9683	796.6659	897.6826

102.581	5562.641	3816.448	135.5294	422.6279	213.0562	1206.241	695.9682
104.1529	5834.83	1179.189	127.8382	504.2653	229.8356	847.8756	827.1966
102.5351	5565.432	4136.568	135.4653	423.1826	213.0406	1241.145	672.9015
102.2253	5577.462	2417.797	135.5524	426.194	213.8436	1026.836	777.076
102.0925	5575.774	3402.328	135.5371	418.9109	213.7163	1136.63	710.0848
104.2509	5815.004	1773.735	127.3278	516.1165	232.1676	931.6749	791.0902
102.0703	5565.245	2456.212	135.3789	429.3038	214.3285	1032.73	772.946
102.1041	5573.435	2540.094	134.863	430.4143	214.4422	1040.272	766.0811
102.8418	5540.732	4362.818	135.4529	417.5039	214.4442	1285.98	670.432
102.7947	5540.348	4321.692	135.4592	417.468	214.4347	1279.925	672.2068
104.0087	5768.118	638.1103	123.9453	478.0951	228.7667	802.9566	879.0919
103.9428	5730.632	246.8127	136.8673	477.8023	202.5799	771.9405	923.4232
102.4416	5578.216	3999.845	135.0693	420.4493	213.928	1216.148	677.2079
102.0784	5572.264	2514.36	135.0012	430.0551	214.4689	1036.91	767.5275
104.1527	5827.862	1200.304	127.7706	505.1313	229.867	853.5697	827.133
101.8906	5584.508	3509.343	135.1087	413.4702	214.1465	1137.913	697.2212
102.1292	5578.685	2211.968	134.1243	430.13	213.5326	1000.451	787.7724
102.2425	5586.341	3038.688	135.9858	416.0212	213.3195	1093.403	736.0825
104.1559	5827.945	1094.382	127.779	504.5561	230.0454	841.1234	834.2713
102.5631	5561.272	3878.788	135.978	421.6282	212.3371	1213.331	692.196
102.8267	5540.589	4351.975	135.4594	417.5668	214.4266	1284.339	670.8424
102.2838	5586.848	2332.376	133.8105	419.5386	214.7297	1012.923	782.7061
103.9342	5729.912	71.1721	136.8927	478.3682	202.2393	751.6547	935.1436
102.8033	5541.068	4497.333	135.4337	416.9021	214.7753	1300.211	660.4864
104.2571	5817.857	1499.566	127.8105	502.9253	229.9659	895.3956	812.0278
104.3488	5828.748	1711.785	127.0032	513.9518	234.9565	920.8044	793.6269
103.8476	5729.046	275.0778	136.9595	478.3906	203.016	773.2	919.4661
102.0649	5565.28	2447.085	135.3843	429.1367	214.3123	1031.445	773.4594
103.741	5730.182	390.4995	137.3331	475.2609	203.7956	782.1531	909.3411
102.5422	5566.272	4095.565	135.4671	423.1557	213.0443	1236.201	675.6142
104.1344	5844.223	981.8354	127.1903	504.9656	227.7476	820.494	838.6246
102.0415	5565.449	2466.957	135.3604	428.4946	214.334	1032.828	771.6684
101.9813	5577.782	3278.571	135.4196	418.5534	214.0711	1117.888	715.3036
102.1317	5578.499	2208.517	134.212	430.5095	213.5432	1000.296	788.0357
103.9564	5744.713	524.3991	123.69	476.6252	227.3163	797.9519	891.4028
103.6704	5727.867	323.4928	137.3003	475.3875	203.5762	773.2623	912.791
103.7234	5722.413	363.6016	137.3079	472.4116	204.0989	781.1337	912.77
102.1413	5580.549	2190.657	134.5187	431.3924	212.9799	997.8437	789.1029
103.9397	5730.636	210.0081	136.8025	477.8061	201.5974	767.5869	926.211
103.7205	5726.533	307.5791	137.3095	474.4772	203.5463	773.2578	915.4312
102.6015	5564.256	3891.563	135.4878	424.4438	213.443	1215.362	690.6057

102.8326	5539.658	4388.031	135.4638	417.4477	214.5288	1289.091	668.7501
104.1104	5824.809	1255.021	127.7815	504.3742	230.6101	859.7943	823.0131
102.1305	5578.567	2242.31	134.1023	430.173	213.5775	1004.092	785.7775
102.1646	5591.49	2854.14	135.7347	412.6834	212.9857	1066.49	746.2008
102.1019	5590.504	2965.153	136.0376	418.7302	211.81	1079.528	737.1001
102.1039	5573.812	2540.229	134.8549	429.8229	214.4152	1039.973	766.0965
102.0886	5575.081	3377.983	135.503	418.8937	213.7212	1133.971	711.7703
102.2439	5590.005	3059.048	136.0282	414.3553	212.5044	1093.84	734.5812
102.0784	5572.32	2531.061	135.0087	430.1224	214.4686	1038.851	766.3913
102.085	5587.226	2612.856	134.4235	434.7746	215.3576	1043.287	756.6831
102.2369	5590.099	3026.683	136.0041	414.1729	212.4995	1089.77	736.5932
102.2364	5590.31	3181.693	135.9058	417.6507	213.1221	1108.6	725.3584
102.0579	5596.944	2920.601	135.8503	412.5472	213.5848	1068.62	737.9659
102.5272	5563.399	3911.402	135.3334	422.9232	213.0018	1215.459	688.2269
102.1181	5591.512	2746.131	135.8038	409.2438	212.9504	1051.621	752.9401
104.1703	5836.471	1304.067	128.7259	506.9274	230.8012	862.8614	818.1072
104.0929	5723.603	40	138.1081	471.9468	203.2999	753.8327	942.8493
102.9268	5539.432	4399.478	135.4571	417.4501	214.4185	1293.357	670.2017
104.124	5829.407	986.5258	127.7395	507.3057	230.0178	827.6399	839.979
102.2662	5577.838	2152.515	134.1941	430.1333	213.5629	997.9901	794.9924
102.0848	5575.826	3442.289	135.6475	416.5399	213.6412	1140.444	707.6677
102.5858	5562.661	3781.761	135.5011	422.6443	213.0635	1202.335	698.3774
104.3217	5819.206	1687.683	127.337	516.5173	234.1693	921.9693	796.6071
104.1493	5832.893	1176.749	126.9442	504.0559	230.4695	848.2798	827.4249
104.7432	5820.795	809.56	128.4057	491.8675	231.8611	825.3595	869.4137
102.5846	5561.415	3922.506	135.4703	422.8682	213.0531	1219.317	689.2031
104.1092	5740.249	453.9326	123.4937	477.5516	227.8035	796.4722	900.1393
102.1052	5579.322	2607.242	134.8745	435.2125	215.0607	1046.766	759.2972
102.1825	5591.226	2873.709	135.7124	412.4088	213.0503	1069.353	745.373
103.7283	5727.498	321.8904	137.204	475.5745	203.9081	775.0089	914.1046
102.0854	5565.219	2456.356	135.3768	429.201	214.3342	1033.186	773.297
102.5989	5559.343	4107.331	135.5158	420.2635	213.0218	1241.555	678.1016
104.0116	5747.341	660.5081	123.3585	477.201	227.4555	814.4995	882.8314
102.8184	5547.106	4274.578	135.4794	420.9144	214.2745	1273.077	673.9096
103.9566	5720.891	86.26269	136.8739	474.11	203.278	756.8995	936.8734
102.2231	5575.145	2316.133	133.7595	427.3945	213.6601	1016.279	784.0927
102.5613	5560.91	4120.89	135.456	421.874	213.6511	1241.713	675.4786
102.0992	5568.31	3382.282	135.4264	418.7458	213.7252	1137.69	713.2082
104.1945	5822.481	1320.974	128.2171	507.2046	231.0546	871.6995	820.3781
104.721	5822.133	757.7812	128.5899	492.0584	231.4212	818.1267	872.2351
104.346	5808.967	1478.699	127.5381	505.3753	229.691	900.1121	817.0387

102.1373	5575.455	2257.672	133.6963	427.2998	213.7006	1006.73	785.9899
104.0527	5728.362	202.4969	137.358	473.5883	203.6004	769.9466	929.6434
102.6577	5563.945	3809.443	135.5572	423.0983	213.4187	1207.264	697.6501
104.5129	5807.773	1588.652	128.1444	506.5432	232.3699	918.648	812.4427
104.1955	5820.826	1296.752	128.3138	506.324	230.8346	869.4056	822.6179
104.3271	5819.997	1683.524	127.4592	514.9218	234.5864	920.8888	796.9346
104.0412	5738.632	582.3007	122.548	477.2456	226.6728	810.1018	890.8766
Jumlah						74	26

Lampiran 3

Tabel Hasil 4.3 Tabel Hasil 2 Fungsi Objective NSGA-II Batara dengan 100 Populasi 75 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif
100.073	6081.321	4800	150.583	1230.756	287.991	1220.02	296.4289
102.4995	6088.63	40	274.2843	25	227.9963	430.7515	898.9171
100	6025.508	928.7245	154.1448	731.8714	285.2258	665.62	649.4429
102.4581	6089.629	996.7284	273.1782	477.9743	204.8126	654.7489	768.0967
100.016	6081.458	3487.678	150.185	1014.451	286.337	1011.138	419.3254
100	6071.574	2616.004	167.9692	1250	288	971.486	440.1572
100	6085.905	2017.415	149.1625	947.3594	285.5782	820.4883	527.467
100	6070.902	2804.692	158.5031	1246.635	284.7536	993.3731	429.1637
100.0289	6037.001	1413.816	155.3064	797.1317	287.6253	734.2978	603.4458
100.0002	6070.37	1713.354	127.1712	725.2166	286.8882	736.9055	586.4863
100.0105	6076.287	2502.183	152.3762	1074.998	288	913.2477	475.3469
100	6081.119	4799.909	150.4565	1163.172	287.6386	1201.028	306.2417
101.8394	6088.259	283.0806	280.5	351.4889	208.3317	521.5401	822.0301
100.0443	6079.771	2025.724	150.2097	992.7531	287.8097	836.6846	520.7964
102.4639	6088.8	489.4451	275.3037	441.8623	210.4445	586.8118	806.0035
100	6081.936	3586.274	145.2775	1033.514	288	1026.829	408.1737
100	6072.245	2267.9	159.6681	1210.617	288	921.014	469.3674
100	6071.065	1729.991	129.5955	750.6443	286.8458	744.8172	581.1232
100	6032.758	1192.172	164.0007	794.0388	287.9985	708.3349	619.2853
100.0538	6085.741	4058.619	157.2615	1076.408	286.9818	1092.259	370.9445
100.1013	6081.273	4752.054	150.201	1130.779	287.494	1190.331	317.1236
100.0711	6081.045	4723.078	150.4699	1224.35	288	1209.512	302.6341
102.3186	6090	837.3308	277.7087	419.185	205.4418	616.9389	785.2678
102.348	6089.342	505.6552	273.4619	410.0509	208.5517	577.1625	808.196
102.4865	6090	1008.071	274.5319	445.6687	209.5315	648.4859	771.3859
102.1721	6089.11	298.9147	280.5	441.4831	206.9156	555.5469	813.8681
102.0717	6089.717	528.0526	265.352	369.4533	200.4816	561.716	810.1296

100.0076	6083.681	4694.852	150.7925	1095.388	285.4575	1171.003	325.0834
100.0477	6079.839	2220.392	149.8183	990.4378	287.7629	858.9047	508.2664
101.9983	6088.74	40	260.6012	297.3531	198.6354	485.1919	853.7744
100	6077.91	2049.172	156.0478	846.6923	283.558	802.3286	545.0459
100.0184	6071.487	2314.559	166.3276	1112.75	285.265	902.7397	484.6181
102.0062	6089.918	781.0637	274.9669	404.1162	198.0208	597.6084	787.4733
102.3336	6089.72	895.4355	273.8037	453.1574	209.5726	632.7722	774.212
100.0027	6078.144	1993.203	155.1828	833.6586	288	792.4019	549.0662
100	6042.535	1282.96	159.2198	796.3477	285.4035	715.5131	611.5717
102.0214	6089.252	787.4193	272.4076	371.5902	202.1891	590.9086	791.0916
100.1513	6071.281	2322.278	165.6739	1226.049	288	935.9737	467.0842
100	6076.038	2483.161	153.7203	1000.605	288	892.1796	488.8997
100	6081.873	3401.819	146.9682	1021.716	288	1002.331	422.5352
101.9984	6088.79	391.0466	280.5	351.0504	212.7345	538.42	816.5337
100.0014	6086.968	3968.154	152.4246	1074.204	286.5749	1079.224	375.8252
100.0385	6084.67	2253.138	157.5424	1037.436	287.9869	871.8385	497.2882
101.9594	6088.214	83.99017	280.5	312.8634	202.7288	492.4632	846.7851
100	6085.762	4052.572	157.5861	1122.603	287.979	1101.435	362.0342
100	6069.748	1782.47	131.9006	766.1606	287.5955	755.2875	575.1268
100	6081.667	3743.531	156.0569	1007.956	288	1038.543	402.5471
100	6085.583	4571.885	151.4206	1060.814	285.4208	1146.943	338.5079
100	6081.277	4564.804	151.3951	1176.982	285.571	1177.02	320.4879
100.0442	6082.251	3863.029	142.2548	996.8511	286.9076	1051.279	397.065
102.3663	6088.992	825.6517	280.3167	433.602	213.4097	620.6936	781.744
102.3313	6089.718	862.8505	273.9063	450.833	209.6295	628.3184	776.7035
100.024	6025.893	976.3564	149.4986	736.2574	288	672.8883	644.6042
100.0002	6085.809	4233.234	154.1421	1084.194	283.1764	1113.17	358.2142
100.0061	6081.039	4606.124	150.9038	1097.72	285.4082	1162.334	331.1762
100.1761	6080.186	1859.026	151.0979	951.0146	287.8176	810.5445	541.7815
100	6087.654	3727.138	152.3131	1083.58	286.1656	1053.128	390.3157
100	6081.119	4025.356	154.7612	1119.843	285.1998	1099.769	366.3315
102.0005	6088.868	397.3214	280.4998	352.3632	213.1345	539.4967	815.7607
100.0061	6079.24	4555.978	150.8699	1095.878	285.586	1156.797	335.1394
102.5097	6090	40	269.9624	200.8232	230.0489	474.4963	868.4754
100	6084.811	2101.99	152.2648	986.2376	287.5179	840.3813	514.9332
100.0002	6041.029	1353.046	161.7574	792.8236	288	723.2962	606.8625
100	6081.62	3723.038	150.6983	1032.807	288	1042.564	399.5135
100	6081.93	3626.928	146.9229	1023.472	287.9046	1029.013	407.2606
100	6034.419	1021.916	154.4306	737.958	287.9996	674.0502	639.1434
100	6080.642	2201.948	146.8206	962.7328	288	848.1384	512.6167
101.9433	6088.215	159.0587	280.5	351.428	212.1244	510.0668	831.0765

100.0069	6071.262	2402.862	165.8165	1210.129	287.6117	937.0774	461.2879
101.9831	6090	163.8228	280.5	319.5565	219.1606	502.839	833.6466
100.0116	6027.515	1205.396	153.0033	750.47	288	701.9723	626.5121
100.1001	6081.283	4711.575	150.1723	1133.609	287.4834	1186.278	319.3249
102.3899	6089.548	203.2674	280.5	153.6124	229.3555	478.0137	863.6765
100.0148	6071.445	2855.556	158.6595	1234.403	284.2537	996.4703	428.2436
100	6070.822	2441.185	162.5712	1212.796	288	942.2953	457.9054
102.5294	6089.158	140.257	269.7573	47.14278	227.3397	448.8261	889.1494
102.5081	6090	154.651	276.742	94.36846	231.1394	460.9494	878.4239
100.0001	6069.036	1803.134	132.0603	769.1634	287.5202	758.757	573.444
102.5296	6089.838	40	272.8333	117.2884	228.8207	454.2264	883.537
100.0101	6079.425	1956.378	157.7429	937.5399	288	813.6554	534.223
102.0082	6088.536	145.7523	280.4637	299.5655	205.415	497.5808	844.7913
100.0187	6071.353	2298.594	164.3716	1108.962	285.2594	900.0633	486.2546
100	6068.8	2607.209	168.5067	1241.309	287.9514	969.4733	442.8402
100	6071.718	2222.813	151.9801	1080.598	287.1271	883.7628	494.1367
100.0014	6087.671	3947.43	152.2483	1071.273	286.6313	1075.773	377.5108
100.0113	6027.847	1191.894	153.0123	751.9912	288	700.6249	627.0808
100.0401	6083.781	2197.439	156.5322	1032.298	287.9731	864.5238	502.0422
100	6076.024	2412.075	153.685	988.8421	287.9797	880.9534	495.6089
100.0374	6079.345	1899.089	160.8985	851.4956	287.8649	786.2146	553.2295
101.9307	6088.238	119.5029	280.5	389.4424	214.2938	514.4934	826.184
100	6086.66	3847.835	159.918	1071.104	288	1064.207	384.1801
100.0011	6030.738	1142.884	153.4817	748.4041	285.3072	692.5254	631.2355
102.3898	6089.283	142.771	280.5	140.3513	226.2199	467.8594	871.2833
100	6071.228	2512.463	168.1365	1241.494	287.8882	957.4279	448.6073
100	6070.148	1870.243	134.5019	795.1422	284.5073	772.6159	565.775
100	6037.059	1324.98	161.9563	794.2694	288	722.0907	609.358
100.0381	6083.182	2198.682	156.9517	1027.24	288	863.5899	502.8976
100.0054	6082.417	4479.912	142.7123	996.7359	285.141	1122.026	355.8072
102.4882	6089.528	40	276.6087	126.6233	229.0703	455.3184	881.2017
100	6081.315	3915.973	157.3623	1061.814	287.1208	1072.264	382.5823
Jumlah						71	29

Lampiran 4

Tabel Hasil 4.4 Tabel Hasil 2 Fungsi Objective NSGA-II Batara dengan 100 Populasi 100 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif
103.9588	6003.821	58.67915	132.4786	269.8161	285.6969	577.4697	876.9479
102.568	6088.464	4800	171.7702	374.8377	223.5247	1079.239	521.5938
103.2191	6089.841	823.2994	170.458	370.6431	212.7976	633.5779	806.16
103.9268	5999.013	501.5911	121.2236	277.6471	278.9027	632.8264	848.7058
102.0826	6088.92	4486.104	166.0108	352.1653	223.024	1022.397	535.1979
102.6506	6088.736	4618.555	168.9991	375.5122	225.9112	1060.612	534.244
103.2277	6089.927	955.8408	170.4803	369.4022	212.8149	648.951	797.7001
103.9207	6005.542	171.455	133.2823	271.5769	288	589.0749	866.9865
102.7154	6087.694	2820.037	166.7625	371.1041	213.4137	852.5685	661.6582
103.8337	5982.259	273.3503	131.67	278.5849	288	610.2463	862.031
102.3632	6088.6	2935.363	166.5908	373.647	209.0481	855.8523	647.2362
103.2063	6087.391	1295.932	168.3357	374.4256	212.0761	690.4381	774.4594
102.724	6089.908	2075.212	168.4831	382.0969	212.9163	767.6801	709.4897
102.0123	6083.717	4071.616	167.1793	355.6449	215.0991	975.2819	565.157
103.2204	6089.838	852.8157	170.4561	369.5229	212.9639	636.7768	804.3399
102.7152	6089.824	2759.715	168.9497	367.7443	209.9106	843.8079	667.3441
102.6582	6090	2662.616	168.4534	367.8671	210.1172	830.7337	672.3655
102.3299	6084.137	3474.297	168.9016	348.925	216.4847	913.1557	612.6611
103.0949	6090	1974.443	171.6463	376.7876	216.1344	765.469	724.2336
102.6015	6083.534	2547.216	172.8065	376.5204	210.3016	820.3819	678.8815
103.218	6090	1550.01	166.2293	376.0759	213.2178	719.7465	756.3634
102.1067	6082.617	3791.033	167.5846	356.6753	215.3268	946.0932	585.9793
102.2607	6081.326	3318.516	164.9565	369.7085	208.3428	899.7236	621.8308
103.8583	5994.99	393.053	122.3942	282.4217	283.6878	620.8454	852.5398
103.1813	6087.651	1222.947	168.3804	374.9905	211.9473	681.2054	778.6685
103.096	6088.568	1620.731	169.3404	384.3553	212.5757	726.9446	748.2566
102.1949	6089.035	4430.644	165.0237	349.6437	217.2364	1018.843	544.1805
102.2496	6082.427	4174.555	167.1918	353.6857	216.342	994.4372	563.722
103.1925	6089.755	1441.035	172.1814	376.147	218.2612	706.0243	761.2914
102.3478	6090	3672.249	169.5716	348.6863	216.4453	934.1668	598.6871
102.2552	6084.248	3533.134	169.0066	348.8599	216.1566	917.7232	607.1873
102.3111	6089.657	3609.338	169.7195	348.5842	216.3849	925.8457	602.1789
103.748	5980.404	357.3378	131.4555	279.6472	287.802	618.5562	854.7996
103.3125	6089.86	1034.781	170.418	368.6663	212.9724	660.5484	794.4153
103.9057	6003.544	151.8263	135.2458	269.164	283.4208	586.6845	870.7987
102.5916	6082.307	2619.382	173.4211	371.473	211.7108	827.7059	674.3978

102.3756	6087.803	3255.237	165.3321	371.9773	208.7001	893.5243	626.7234
102.0414	6085.454	3872.406	168.1306	350.1857	216.3683	950.7218	579.1444
102.3451	6090	3773.274	169.6758	347.784	215.5515	945.6746	592.4207
102.041	6085.352	3905.095	168.1197	352.1013	216.3633	955.0469	576.6603
103.1264	6089.854	1812.403	169.5655	375.6745	217.2649	747.3252	735.3866
103.1455	6088.333	1372.636	173.422	369.4222	212.592	695.7159	768.6459
103.1078	6087.993	1872.743	173.7812	378.9339	212.3852	755.452	733.0507
103.8714	5999.842	74.02996	129.8215	270.7594	288	578.6104	873.5704
103.9278	5998.492	499.7488	121.4443	276.9293	278.808	632.6813	849.1324
102.5985	6082.811	2493.631	172.9275	376.9218	210.4712	814.4431	682.4107
102.2538	6081.41	3379.997	165.3931	369.5793	207.9829	906.6182	617.7521
102.1861	6082.816	4281.592	166.8971	351.6248	215.9159	1004.361	555.5798
102.1131	6083.695	3974.187	167.2761	356.1199	214.2659	967.0928	574.1985
102.7155	6087.85	2772.371	166.8069	371.2197	213.8697	846.9564	664.5984
103.0589	6088.601	1675.486	169.4858	383.1486	212.1878	731.9126	744.1348
102.3669	6087.549	3189.688	165.6193	371.8249	208.5588	885.6842	631.0519
103.1892	6089.973	1380.351	171.955	374.3494	213.7821	698.4584	767.3699
102.1349	6085.887	4369.24	165.9607	352.2843	216.0185	1011.912	547.7199
103.2352	6087.962	1111.091	168.3149	374.4709	211.9449	669.5077	787.3535
103.2164	6090	1090.816	171.1532	374.0573	210.257	665.5542	788.7554
102.6071	6089.972	2300.977	168.0363	382.7962	212.778	790.6792	691.7134
102.3361	6084.374	3573.813	168.9975	348.1755	216.2785	924.6666	606.3312
103.1387	6089.707	1733.551	171.155	382.3374	215.7994	740.2202	740.5317
103.8777	5983.899	280.9723	130.2266	283.5047	288	613.0233	861.2657
102.3373	6087.485	3087.223	167.5738	370.38	201.7904	872.667	640.388
102.1808	6082.809	4243.107	167.0157	352.0401	215.8541	999.8173	557.9906
103.1673	6089.295	1527.315	169.2331	374.7859	212.6728	715.4768	757.4792
102.3833	6083.478	3085.579	166.4369	371.5998	208.9737	875.6946	639.1475
102.3952	6086.587	3029.295	165.2208	368.5918	207.0234	867.4934	643.7484
102.6194	6089.852	2209.783	168.0568	383.2786	212.9841	780.5735	697.9302
103.314	6089.864	1050.571	170.4178	369.2249	212.9535	662.5741	793.3103
102.3117	6089.705	3721.891	169.734	349.1866	216.9526	939.1078	594.3442
102.3729	6086.739	2978.571	165.2063	368.8942	206.8984	860.9194	646.5952
103.1393	6089.688	1769.663	171.1347	382.6352	215.8126	744.5358	738.0869
102.6387	6089.856	2181.591	168.0655	383.3556	212.9827	777.879	700.2299
102.3421	6084.119	3394.07	168.9345	348.1533	216.6171	903.9739	618.365
102.6619	6090	2249.837	168.2646	382.0409	212.2143	786.1668	696.7217
103.1982	6087.806	1191.966	168.3831	374.3782	211.954	677.8753	781.1786
102.0862	6083.27	4036.957	167.2463	355.9434	215.0923	973.7215	569.1834
102.0807	6083.635	4009.744	167.3886	357.0361	215.149	970.4894	570.5958
102.1882	6085.826	4407.949	165.9975	349.5186	215.9865	1017.363	546.8285

102.62	6089.848	2268.863	168.0391	383.3287	213.0324	787.4951	693.9747
103.219	6089.968	1154.723	171.9746	374.3644	213.894	673.0295	783.0343
102.6808	6089.088	4764.88	172.3699	372.7994	224.8155	1077.683	526.1697
102.5935	6090	4708.356	172.0633	376.8174	225.735	1069.057	526.7065
102.6784	6089.999	2347.092	168.2514	382.0251	212.2457	798.0034	690.5978
102.3103	6089.649	3707.129	169.7337	348.6479	216.3613	937.2506	595.6433
102.3826	6083.464	3143.846	166.2322	371.2753	208.6405	882.4147	635.4332
102.3833	6083.465	3128.892	166.4584	371.7404	208.9644	880.7881	636.2442
102.3878	6086.4	3008.533	165.4562	370.2929	206.5017	865.3652	644.9524
103.1648	6088.894	1484.144	169.5031	373.8423	214.6366	710.2288	759.7405
102.6798	6090	2427.757	167.6217	376.8175	211.9378	806.1834	686.2147
102.1349	6085.904	4325.027	165.9625	352.3031	215.9235	1006.754	550.7003
102.689	6090	2384.002	168.0394	381.8751	212.3557	802.5939	688.3447
102.6162	6088.388	4747.326	171.8428	374.7579	223.4166	1074.554	526.2683
102.1305	6085.863	4326.177	165.9826	352.3087	215.9203	1006.777	550.5358
103.0815	6089.155	1894.683	170.5241	380.6639	213.8424	757.2112	729.6858
103.0889	6089.748	1947.157	170.1583	377.4529	215.1261	762.4656	726.2081
102.7041	6090	2145.833	168.478	384.1747	214.0251	775.7647	703.505
102.112	6083.876	3979.222	167.198	356.0171	215.1419	967.5148	573.4457
102.6879	6089.985	2121.621	168.3789	382.6984	212.9775	772.1277	705.4341
102.04	6085.445	3890.778	168.0977	352.1366	216.3626	953.3167	577.566
103.1221	6089.629	1741.035	171.1752	382.463	215.7226	740.6635	739.689
102.364	6087.586	3221.456	165.3995	371.9671	208.7692	889.3218	628.7373
Jumlah						70	30

Lampiran 5 Tabel Hasil 4.5 Tabel Hasil 5 Fungsi Objective NSGA-II Batara dengan 100 Populasi 100 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif	F(3) Leadership	F(4) Egois 1	F(5) Egois 2
100.0002	5630.668	4800	3.4605	748.8355	253.162	1298.704	479.9091	751.877	-1084.19	-1999.93
100	5836.834	4800	70.37586	688.9257	249.357	1192.279	450.1517	726.3791	-927.07	-1685.83
100.0003	6090	40	120.3125	580.1518	70.07802	504.3666	807.964	-111.626	-829.838	-1656.01
106.8651	6090	702.1558	3.4605	1238.505	244.7629	950.2227	729.9248	-49.7517	-1335.82	-2640.08
107.7015	6084.039	40	6.260407	1180.77	228.1525	886.6728	810.8624	-134.39	-1292.22	-2575.49
100	6089.978	40	76.4935	958.5491	73.43568	600.3157	741.3063	-202.095	-1102.03	-2200.28
100	5626.35	4800	3.570651	607.2477	254.6037	1265.121	503.8432	783.7733	-1034.83	-1901.17
100	5749.145	3167.136	94.03283	578.1793	241.5879	1011.56	600.8863	504.2363	-774.761	-1435.9
100.2596	5758.958	3160.261	87.99828	578.4088	241.8181	1014.544	604.6228	503.2934	-795.148	-1476.89
107.3968	6069.055	2937.03	21.13851	1246.815	224.3416	1238.153	605.4418	335.9722	-1262.96	-2420.55
104.744	6052.684	1332.615	41.82168	1050.546	282.1091	926.7999	665.938	84.34612	-1161.57	-2269.31
107.1872	6069.862	2686.528	19.11542	1232.05	219.6782	1198.825	621.556	293.2834	-1268.27	-2439.66
100	5837.133	4498.777	84.20904	672.4357	247.7416	1152.477	474.2815	682.3053	-875.587	-1592.96
100.0186	6090	1141.547	116.0722	779.1941	85.76383	682.8081	695.0177	29.57233	-910.046	-1779.18
106.3126	6078.155	2157.28	32.24572	1198.552	238.474	1097.806	633.7697	201.332	-1229.91	-2379.96
107.1456	6090	886.2301	3.4605	1250	244.6366	982.9915	722.1017	-17.3659	-1335.45	-2633.2
100	6090	122.6127	91.89484	794.2458	90.33355	567.792	756.9073	-148.244	-995.932	-1984.77
106.7811	6073.659	2324.803	14.91692	1250	227.3776	1147.166	629.3012	221.8678	-1295.92	-2506.76
100.1156	5601.585	3779.141	4.104579	652.8128	252.882	1171.652	573.0178	611.6467	-1036.18	-1937.96
104.9523	6089.067	1997.936	34.35869	1075.167	285.067	1001.209	612.6776	182.6807	-1206.69	-2337.28
100.013	6089.131	50.739	79.79782	956.0998	80.90848	601.355	738.5268	-198.084	-1089.65	-2174.91
100.1623	6073.022	1811.619	116.9018	747.3223	79.00725	764.8829	665.4281	154.1604	-887.426	-1711.83
100.1509	5619.042	4665.427	3.542267	686.8818	225.8078	1277.985	516.5188	745.7762	-1056.16	-1949.27

100	5778.9	3540.801	65.48512	654.4676	232.3053	1062.594	559.2541	536.0632	-907.747	-1689.72
100.0019	6090	692.0447	87.02028	878.4212	99.90434	655.1693	700.7266	-71.7263	-1040.21	-2054.02
100.0235	6084.437	40	116.7729	613.5766	93.458	515.1702	794.2035	-115.888	-850.111	-1695.77
100.1553	6076.438	1508.005	111.3804	792.0493	81.53728	739.0532	675.9872	92.01513	-922.267	-1791.55
100.0024	6089.408	40	97.77699	755.3031	103.0967	548.1265	764.0646	-151.151	-963.061	-1921.35
100	6089.526	616.9029	90.71656	867.2854	93.36724	643.8628	710.5612	-81.5522	-1024.01	-2024.34
100.0019	6090	520.5745	87.43904	862.6938	98.58169	631.2635	715.3517	-96.9231	-1033.57	-2046.51
107.3167	6070.196	544.7608	6.118997	1250	228.2702	957.3266	759.9603	-67.5914	-1315.77	-2605.77
100	6090	40	94.02353	775.0073	64.83	554.1234	776.3541	-159.544	-982.424	-1961.35
100.0121	6079.819	294.9758	105.0697	760.8822	96.24832	583.699	751.6497	-106.434	-936.471	-1859.9
106.7134	6089.767	160.8258	16.04624	1167.972	236.2079	864.8514	778.6014	-125.103	-1272.54	-2531.84
100	6089.996	164.6339	78.99282	896.8627	86.24389	598.9168	738.0629	-166.395	-1073.14	-2137.92
100	6090	547.1841	82.15631	856.3758	102.0657	632.7914	712.8722	-91.914	-1049.1	-2076.57
104.5232	6080.864	1381.83	41.94539	1031.931	280.2565	909.11	655.4641	86.09535	-1170	-2284.61
100	5803.9	4276.657	80.1535	665.4455	243.7921	1139.483	498.898	653.9682	-872.928	-1595.17
100.0803	5843.237	4533.613	77.51336	678.2776	248.9385	1157.948	470.6355	685.2842	-901.193	-1642.97
100	6090	40	93.71132	791.3019	77.88998	557.7721	768.1811	-162.049	-988.896	-1973.86
100.2202	6090	668.0224	73.84851	842.7582	70.55691	651.4183	724.7539	-69.808	-1068.95	-2113.29
100.1912	5651.369	4249.486	3.497656	675.3986	217.9258	1214.053	543.3464	670.7975	-1065.34	-1981.77
100	5609.647	3612.955	6.226393	634.3236	248.3563	1140.759	584.8218	584.2524	-1028.04	-1927.37
100.042	5637.857	4684.342	3.707271	613.2071	257.0225	1249.307	508.0119	761.1882	-1040.52	-1916.33
106.6176	6071.001	1744.287	24.2513	1249.493	233.48	1075.044	662.9073	125.8118	-1265.98	-2466.03
100	5630.975	4746.782	3.4605	716.0941	254.5885	1284.124	488.2488	750.1408	-1073.4	-1980.09
104.9971	6087.015	1820.41	36.33028	1050.034	288	976.2864	629.0326	160.3636	-1190.21	-2310.14
105.7037	6090	42.31191	21.00044	1122.537	223.9403	809.5166	776.6644	-150.247	-1256.12	-2503.37
106.645	6085.741	60.97092	11.85955	1173.91	237.1203	854.4881	783.0127	-143.813	-1287.82	-2565.71

106.633	6090	40	13.05857	1100.24	231.318	831.5698	797.9721	-132.836	-1261.22	-2513.4
106.8205	6090	1413.921	28.58247	1250	221.4363	1034.74	690.5316	68.79787	-1256.58	-2458.66
100.2576	5757.062	3230.305	89.01582	578.7311	247.8429	1023.324	597.8066	516.0477	-791.103	-1466.27
100.0011	6089.991	128.2094	115.3288	630.7657	83.06437	527.0739	788.0088	-107.791	-863.305	-1719.57
100	5619.658	4467.477	5.773159	629.2797	223.3521	1235.704	536.9215	723.0887	-1032.04	-1907.72
105.8271	6090	1152.461	32.90124	1131.351	287.9704	942.4082	677.8775	42.4347	-1217.54	-2387.07
100.001	5772.768	3869.211	58.99071	705.5781	231.2625	1116.626	530.3033	579.8868	-943.861	-1751.04
100.1683	5615.804	3918.328	3.4605	675.595	244.3801	1189.31	561.5568	626.3067	-1051.06	-1963.36
100	5625.599	3946.975	15.89241	669.0079	251.2256	1181.069	552.6044	630.5916	-1014.03	-1888.12
100	6089.943	1417.39	133.4825	787.9427	86.18072	716.0506	675.4614	76.90339	-855.182	-1660.24
105.1006	6090	1808.742	36.80246	1077.718	283.8882	983.7775	628.6141	152.978	-1197.56	-2325.36
100	5808.11	4310.096	78.28363	678.9832	243.1128	1145.03	493.6898	655.125	-885.428	-1619.08
100.0124	6065.451	1052.883	116.4561	836.6902	80.78735	697.4432	698.6182	8.043945	-917.795	-1797.8
106.5717	6081.525	2161.089	28.40234	1182.57	228.9024	1101.017	645.0764	206.9081	-1234.91	-2390.15
100.012	6085.791	317.6266	99.12622	761.4528	101.8522	583.9056	746.117	-104.998	-958.963	-1903.94
100.0002	5637.653	4800	5.479646	751.064	256.8856	1296.043	476.5737	750.3621	-1081.11	-1993.65
100	5840.451	4361.699	83.21236	681.4158	248.1246	1137.312	480.9951	656.5165	-883.285	-1612.91
107.3141	6070.196	541.421	6.178378	1249.909	228.2745	956.8329	760.1399	-68.1557	-1315.58	-2605.5
100.0005	5838.317	4685.956	69.90692	684.1162	247.2994	1177.227	459.0791	707.7848	-927.641	-1690.84
105.7613	6082.108	985.3454	25.49852	1178.272	218.4497	938.6532	710.026	-1.87107	-1255.55	-2470.98
107.6246	6084.209	47.05353	5.175242	1187.324	228.8986	886.7664	807.1689	-135.984	-1299.24	-2589.28
100.0461	6067.038	1844.452	106.0742	812.777	90.5654	784.1579	645.6534	144.0005	-944.586	-1824.67
100	5843.73	4548.62	77.6804	678.2982	248.2598	1157.099	468.0098	686.4305	-902.052	-1644.21
100.0239	6090	205.0038	112.9925	613.3968	89.20294	532.2437	783.6199	-90.8446	-864.965	-1720.12
100	5595.572	3496.026	5.318008	639.2177	247.54	1134.498	595.1457	566.9729	-1026.83	-1928.88
100.1839	5839.255	4641.861	71.56011	690.558	247.6896	1178.72	464.7887	701.9196	-921.917	-1680.85

100	6090	60.98525	117.6141	581.7904	63.70182	507.5187	808.8075	-109.559	-839.383	-1674.61
105.0148	6090	1254.773	38.01635	1034.238	280.453	905.7793	672.3549	68.74107	-1180.3	-2309.43
104.933	6081.804	1249.813	39.31177	1040.144	280.3355	907.7349	671.7497	67.70563	-1175.77	-2300.53
100	6087.895	1247.863	125.1769	799.1066	83.0163	700.3331	686.2495	44.81702	-885.736	-1727.11
100.1956	5748.966	3191.146	88.48196	578.479	241.7578	1020.561	603.3266	510.0557	-790.355	-1466.28
100.0004	6090	1485.761	126.9324	793.2118	80.65671	725.7331	671.9951	85.37768	-878.79	-1705.37
105.8051	6081.016	163.3828	23.97904	1137.262	234.5291	833.8062	766.1041	-128.014	-1245.84	-2478.41
105.9017	6077.947	759.4802	24.58241	1203.661	217.7735	924.7177	725.1487	-42.1095	-1263.23	-2493.89
104.9357	6090	2054.355	35.11873	1070.341	284.0521	1005.691	609.6065	192.7152	-1203.18	-2328.42
106.6413	6090	70.0636	8.202322	1143.068	243.4554	845.7919	783.7177	-136.939	-1291.56	-2572.67
100.0473	6066.786	1833.668	107.088	808.2167	90.90909	781.8606	647.1223	143.504	-939.563	-1814.97
100	5784.851	3932.894	57.75085	703.8131	230.9379	1118.399	523.784	587.5718	-952.456	-1766.12
104.7905	6080.721	1517.884	41.52912	1042.717	281.7707	935.7259	649.9911	110.5232	-1170.92	-2281.85
106.3222	6078.306	2161.905	27.02739	1234.949	245.4934	1107.607	624.3926	193.8642	-1259.35	-2438.46
100.0681	5637.964	4753.678	3.869149	618.8366	254.391	1259.62	504.1183	771.7013	-1041.51	-1916.09
100.0748	5605.612	4162.115	3.778643	678.2547	218.7996	1220.87	555.6394	665.4386	-1048.04	-1950.05
100	5607.468	3459.191	5.313941	634.6681	246.059	1123.957	596.3989	558.721	-1030.29	-1937.07
100.0583	6090	1357.729	106.2767	787.5971	76.47837	711.9578	683.4779	61.64498	-944.902	-1842
100.1222	6090	1318.314	110.3922	784.1067	88.73554	707.8578	683.2235	58.63499	-929.062	-1811.22
105.9619	6083.057	744.6452	25.08493	1198.08	215.3678	921.2469	728.3426	-43.8473	-1260.92	-2489.84
104.9999	6089.089	2207.102	35.73523	1071.64	282.6793	1026.184	601.4527	219.092	-1200.22	-2317.44
100	5778.11	3900.547	60.71973	702.5531	230.7161	1117.14	527.8519	584.7141	-939.331	-1740.95
106.3058	6088.771	1990.808	26.36394	1169.062	229.5453	1066.702	650.7562	175.2999	-1244.21	-2414.41
105.7734	6071.073	701.7189	25.12299	1248.215	218.6515	928.1983	719.8361	-61.4155	-1275.34	-2520
100.0216	6084.709	649.7993	73.20256	879.0555	90.39457	654.0622	708.2996	-80.8415	-1083.98	-2143.29
Jumlah						78	22	0	0	0

Lampiran 6

Tabel Hasil 4.6 Tabel Hasil 2 Fungsi Objective NSGA-II Wira dengan 100 Populasi 20 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif
50	5413.911	312.9575	203.5902	67.02311	81.21048	-802.254	107.0942
50.04872	5461.202	1110.984	206.2488	134.4907	91.35264	-711.742	14.46782
52.19294	5486.046	420.0994	72.35952	675.7439	182.6919	-602.052	-14.3172
50	5503.105	793.2134	213.139	85.50349	78.01118	-780.466	42.83302
50	5428.995	369.1007	192.6023	83.86154	82.42755	-797.705	93.87382
51.65069	5769.973	5200	243.9575	529.8822	287.7732	-229.388	-479.83
50.00106	5414.16	322.1242	201.0495	72.72918	82.97736	-799.808	104.4629
55.75935	6090	4706.223	64.16243	173.4588	21.38044	-376.647	-275.526
55.76681	6090	4325.972	75.31374	118.3646	13.79507	-434.678	-229.584
52.28392	5483.808	253.4265	71.93776	675.2681	181.337	-617.858	3.252747
50.01031	5462.575	1021.005	209.8391	65.22044	84.14356	-741.184	36.48583
55.76435	6090	4690.652	64.30833	173.4673	21.67154	-378.326	-274.138
51.75145	5805.237	4969.538	231.8287	448.6889	287.862	-288.431	-451.414
51.61699	5772.176	4599.812	210.9848	467.1137	286.5076	-315.926	-421.803
55.97837	6090	4271.338	75.38759	187.6154	18.30106	-417.545	-233.653
51.73385	5782.454	4768.633	231.6979	472.9591	294.9	-296.688	-436.131
50.0721	5490.592	4967.107	268.4611	841.3171	262.3777	-104.963	-491.422
50	5451.576	30	199.6389	89.88591	76.27579	-845.575	117.5238
55.70003	6088.446	4516.617	66.17913	172.3265	34.04638	-400.646	-263.854
50	5428.686	372.9838	192.3649	83.84326	82.60385	-797.121	93.56336
50.10253	5491.733	871.3819	210.1135	88.60925	74.01704	-762.333	42.31272
51.56046	5441.648	543.9021	75.69078	729.0777	163.1641	-573.471	-37.9609
50	5436.638	474.942	207.7348	86.29662	100.3192	-789.161	79.69632
51.4917	5781.184	4709.878	210.9847	451.9543	294.9	-314.817	-436.052
51.7661	5479.552	935.5186	78.69313	729.9317	169.3804	-538.132	-75.3269
55.30292	6032.321	4919.571	59.32097	163.0236	27.61451	-343.11	-292.533
50.11134	5519.558	5098.892	265.3619	876.7193	268.3155	-92.2069	-517.344
52.3312	5496.059	375.3405	74.73898	687.5635	166.0104	-604.034	-6.89872
50.01979	5462.21	1038.802	208.288	97.30849	90.60241	-730.806	27.19076
51.64337	5776.841	5159.841	235.3352	466.3693	294.9	-253.097	-468.361
51.65957	5779.735	4458.234	230.1635	458.138	294.9	-337.605	-409.002
51.60647	5772.235	4606.347	210.9605	467.0506	286.5627	-315.522	-422.682
52.21192	5500.613	516.3712	72.11681	689.5753	170.6523	-592.697	-25.1931
55.7142	6080.796	4418.006	64.64108	172.3616	9.739799	-407.541	-247.383
50.04872	5461.143	1113.037	206.2967	134.8895	92.66298	-711.422	13.90894
51.75187	5811.019	5066.778	230.5908	471.3065	293.8173	-274.082	-467.087

50.07267	5437.151	172.3135	207.3144	97.44888	65.19399	-818.537	113.4761
55.57501	6080.575	4498.011	65.59235	170.3649	19.55645	-403.145	-260.131
50.09905	5504.388	5043.591	272.9057	850.4802	279.7059	-99.6444	-506.156
55.80732	6078.552	4377.04	66.81111	131.9697	4.005	-418.533	-230.886
50.04979	5433.721	267.5439	207.1563	83.9144	58.80552	-809.792	109.9995
55.99359	6090	4308.623	61.14331	186.9218	15.9447	-412.359	-237
51.56206	5436.836	531.8141	73.60846	720.2898	169.5575	-575.089	-35.7517
55.63069	6090	4783.847	61.6308	161.7678	18.39165	-374.189	-283.022
55.68797	6090	4771.887	62.02551	173.5728	22.06317	-371.051	-283.546
55.78434	6084.18	4428.646	75.98198	180.2984	11.73288	-404.122	-247.963
50	5440.922	1239.107	206.1325	134.8754	92.74256	-689.414	6.963204
50.0503	5434.256	236.4674	207.9161	93.27415	51.24489	-811.067	112.565
50.06453	5432.248	295.5079	207.5347	113.8992	60.22279	-798.013	102.1663
51.75195	5474.378	908.262	77.09992	706.0683	161.0215	-545.128	-65.4834
52.25405	5493.513	349.5937	75.46716	711.4781	168.6821	-602.384	-11.8089
55.59278	6088.101	4564.889	64.28556	171.7652	15.55777	-397.544	-266.463
51.76395	5792.271	5166.628	227.6914	481.3048	293.1122	-251.326	-472.471
50.0089	5463.697	1109.278	208.2597	98.77157	91.43225	-723.216	20.11643
50.01044	5463.679	991.4409	209.8281	65.42508	84.16397	-745.057	38.63002
50.0686	5432.203	374.3705	207.8039	107.632	90.61648	-791.259	89.40949
51.61051	5772.273	4595.431	211.0622	467.0779	286.5193	-316.686	-421.644
51.74033	5782.358	4773.014	231.6205	472.9949	294.8883	-295.928	-436.29
51.74406	5800.532	4850.228	230.9214	484.9354	289.6844	-291.502	-448.313
50.10237	5433.686	40.13716	198.1071	68.28742	79.31873	-839.019	127.6284
50.13902	5493.001	5200	264.5904	851.5548	272.1143	-74.4643	-514.293
56.08065	6074.745	4261.74	71.89183	186.299	15.0932	-409.092	-225.256
51.76948	5472.943	948.6481	77.4798	735.7514	163.273	-531.936	-74.4258
51.67701	5771.285	5181.251	244.2054	545.4623	280.4399	-227.223	-479.064
51.68475	5765.196	4422.651	230.4156	468.0046	283.1766	-331.852	-400.664
50.14617	5484.822	4961.371	270.0333	877.1775	261.2502	-91.9592	-494.012
55.47856	6020.349	4919.32	61.12156	165.9746	29.24994	-332.059	-285.069
52.31686	5475.194	287.4611	73.34	677.5881	170.0575	-608.257	6.054199
50.11968	5461.857	1132.55	206.4156	138.8439	89.63854	-706.241	14.20996
51.56545	5436.619	546.9544	73.58894	720.2723	169.3355	-573.123	-36.8003
51.76609	5473.159	933.5078	77.49931	735.7689	163.495	-533.901	-73.3772
50	5442.388	959.2685	193.9626	112.3555	90.64302	-727.852	33.7283
55.92682	6090	4697.331	67.35855	171.3403	36.0379	-373.785	-272.681
51.67925	5764.454	4607.412	230.7205	465.0111	287.3296	-311.037	-416.45
55.59905	6088.138	4565.321	64.19437	172.3725	15.55023	-397.166	-266.449
50.10317	5490.376	862.5957	210.0915	88.7711	74.1215	-762.712	43.34281
52.13086	5471.655	515.6194	73.7909	716.9845	161.3177	-575.56	-23.3033

51.74781	5784.554	5176.41	231.1903	469.9918	294.9	-250.33	-469.676
52.22525	5486.029	459.9243	71.94275	660.6038	182.635	-600.199	-13.6619
50.04676	5446.893	593.6072	207.6872	84.71625	97.17771	-778.647	69.74331
51.84726	5793.055	4952.851	226.2398	459.2203	290.2313	-279.484	-447.362
56.20472	6089.979	4850.598	68.30706	174.5337	42.53625	-347.007	-279.28
55.85045	6090	4445.96	80.98591	173.9144	11.92126	-404.41	-247.148
50.01113	5461.477	710.1213	211.4471	93.38642	86.78036	-770.053	56.56032
50.01314	5463.688	1109.272	208.2571	98.81355	91.46114	-723.076	20.23051
50.04374	5432.725	521.8197	207.4055	66.76192	106.2591	-785.754	80.46949
51.76047	5765.475	4721.545	232.8083	462.7662	290.3049	-296.457	-423.865
51.75168	5775.582	4810.046	233.5703	477.7249	287.3944	-286.964	-436.219
50.12642	5441.25	107.4476	196.7091	70.29504	75.3125	-833.04	121.31
51.79902	5767.664	5022.613	230.1953	465.5886	294.9	-260.485	-450.319
52.32706	5493.271	369.0562	72.56352	689.5644	164.8419	-603.071	-6.12756
50	5415.414	383.4875	203.5547	68.54782	82.04453	-794.322	100.324
50	5406.38	145.0745	201.9268	81.12412	81.1919	-814.995	119.9855
51.77921	5773.103	4854.671	231.8942	426.7136	294.9	-292.804	-430.333
50.12165	5491.527	862.7617	211.3611	71.22184	71.93146	-766.994	47.77986
51.71395	5780.331	5107.391	229.4138	464.928	294.9	-258.775	-463.049
50.27334	5469.182	1182.891	189.7477	100.9185	86.33951	-707.748	19.53679
55.43761	6021.248	4788.174	55.26153	190.2666	28.51056	-342.684	-280.853
52.29291	5471.232	382.0382	69.57582	639.031	169.6303	-605.725	5.886417
50.13956	5492.94	5094.659	264.4031	864.0853	265.2486	-83.3439	-506.292
Jumlah						29	71

Lampiran 7

Tabel Hasil 4.7 Tabel Hasil 2 Fungsi Objective NSGA-II Wira dengan 100 Populasi 50 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif
51.80149	6088.781	30	85.95863	316.7125	129.6195	-1008.93	-57.8021
50.00729	6005.36	5198.927	127.9846	597.4005	51.39616	-352.192	-553.896
52.03068	6088.544	1335.682	92.89859	492.7909	158.874	-806.814	-201.509
50	5980.861	1908.348	124.7664	570.0602	43.67613	-732.166	-266.699
51.88766	6090	667.001	88.53731	479.5161	145.5631	-892.478	-144.894
51.75741	6086.073	167.3526	86.71305	480.9644	131.3159	-952.078	-103.093
52.00187	6086.678	567.6687	90.5521	460.8322	139.8381	-903.748	-126.99
51.74162	6083.447	49.22467	87.89721	481.5295	127.4309	-964.964	-92.0892
50.09551	5984.475	3845.567	124.787	611.9264	58.6159	-494.889	-438.278
52.04007	6086.35	375.3072	90.39434	439.7822	141.8653	-930.227	-106.044

50	5984.033	3937.042	128.5578	603.527	62.81413	-489.257	-447.648
51.75583	6086.816	1267.475	90.35772	489.1113	145.4131	-822.655	-199.792
50	5986.134	3757.577	124.3397	612.6009	58.60171	-508.555	-434.402
51.86265	6090	77.00548	87.53551	339.6518	123.0024	-996.242	-62.9651
52.01723	6088.086	252.777	85.99762	343.3979	124.475	-969.329	-73.7678
51.73109	6087.398	1212.81	88.68807	481.575	144.2055	-831.814	-194.482
51.74015	6087.398	155.9655	86.67323	327.4598	121.8381	-992.558	-69.9263
50.04107	6005.585	5164.617	128.2483	596.4637	51.51984	-355.526	-549.897
50	5972.544	3491.914	119.9505	622.6685	64.51461	-531.194	-412.796
50	5982.313	2253.32	125.6414	567.9676	60.40153	-693.658	-299.485
50	5993.443	2325.887	128.6862	577.5743	46.42318	-687.248	-306.437
51.74229	6083.423	89.82346	87.73881	482.3026	127.6753	-960.007	-95.6777
50	5978.612	3422.088	120.983	625.1116	60.79905	-541.27	-407.951
51.71813	6088.249	913.0458	88.96713	476.4201	140.5196	-868.72	-168.123
51.86741	6090	752.1846	93.22788	468.085	147.0128	-886.21	-150.207
52.00288	6088.809	211.5827	82.15359	338.3275	123.3369	-975.981	-70.0321
50	5977.135	3707.277	127.535	608.5293	51.34366	-511.406	-425.012
50	5969.202	3613.179	125.0211	598.8607	55.86447	-521.431	-414.635
51.83524	6090	1001.961	88.65427	476.3396	144.5182	-855.735	-173.472
51.74479	6088.614	878.896	89.722	476.7824	145.2528	-872.155	-165.749
50.00884	5981.461	2532.281	125.1159	574.3045	52.16649	-658.602	-321.514
50	5977.542	3682.328	127.0755	597.1711	55.04527	-517.441	-421.734
50.05869	6003.027	5074.25	127.7644	595.8998	50.62581	-364.526	-540.91
50.03111	5994.276	4458.862	130.3571	610.7889	56.21628	-429.907	-492.384
50.06393	5981.02	2754.974	125.5884	622.3209	67.23393	-619.291	-351.632
50	5989.78	4809.511	126.483	603.007	57.99336	-389.859	-520.689
50	5983.25	4132.493	128.3571	599.12	62.6941	-467.206	-462.849
50.00629	6001.567	4907.85	128.8478	595.4425	50.76733	-385.034	-528.086
51.88609	6090	1059.016	86.29388	481.8343	145.2297	-846.124	-178.214
51.99806	6088.373	824.7265	87.57823	469.9041	144.1415	-872.383	-152.137
50	5983.516	3984.409	124.586	617.5186	57.88093	-479.712	-453.428
50	5993.442	2375.926	128.5515	579.0036	46.45599	-681.05	-310.914
51.95799	6088.952	1163.085	89.3396	478.0935	148.501	-832.517	-184.232
50.03628	5980.177	2890.976	122.2396	610.2923	64.22505	-606.683	-360.761
50.01153	5988.98	4775.504	127.0839	605.6555	53.96854	-392.358	-516.773
50.0311	5994.22	4414.315	130.304	610.7992	56.37822	-435.082	-488.705
50	5983.639	4197.458	128.3936	598.8182	62.94037	-459.88	-468.357
50	5993.391	2454.154	128.6639	577.1033	46.34071	-672.376	-317
50.00037	5984.147	4197.793	126.5659	615.2889	52.61252	-455.527	-469.396
50.01144	5988.93	4656.975	127.1918	605.5243	54.11932	-406.209	-506.886
50.0554	6003.036	4988.412	127.8062	596.3821	50.78261	-374.529	-533.989

50.00947	5971.656	2620.196	126.3959	568.9315	54.18502	-645.53	-325.673
50	5980.272	2550.999	124.8997	574.2995	50.86687	-656.119	-322.738
50.03023	6003.908	4956.792	126.1023	588.3299	49.51344	-381.266	-530.57
50	5979.071	3094.872	122.0048	607.443	71.04301	-584.436	-379.723
50.00376	5993.027	4327.454	127.0734	615.6992	64.2894	-444.45	-485.27
50	5986.434	4056.662	128.4714	624.6618	53.45196	-470.743	-460.112
50	5974.54	2969.651	124.5796	599.9906	58.96241	-598.628	-363.387
50.0122	5988.095	4711.25	127.3637	609.205	53.86774	-398.569	-511.834
50	5993.384	4282.757	127.9979	611.3104	57.62366	-450.838	-479.111
50	5981.655	2203.145	126.5099	568.3758	54.60031	-698.96	-293.684
50	6007.255	5016.882	128.9547	601.1146	49.08893	-373.497	-539.487
52.01983	6090	811.2633	93.18975	468.9506	149.6975	-874.616	-151.406
50	5973.572	2756.848	125.5871	599.2478	62.20756	-623.363	-345.973
51.96774	6079.231	431.705	86.81405	434.265	152.216	-924.337	-112.977
50	5993.422	2416.346	128.5389	578.479	46.19027	-676.447	-314.107
51.80651	6090	1022.537	88.57574	476.5641	144.8944	-854.15	-176.195
50.01964	5983.171	2839.735	121.7763	606.5997	63.12151	-615.329	-356.771
50	5989.595	2084.109	124.0641	582.041	40.68625	-712.326	-285.248
50	5973.694	2655.239	125.3957	602.3223	66.44834	-634.637	-339.23
50.03211	5995.522	4586.6	130.4403	610.7142	56.22187	-415.537	-503.288
50	5973.563	2679.735	125.3169	602.3886	66.02647	-631.689	-341.155
51.97127	6080.377	463.6426	87.38731	445.8872	145.6001	-917.895	-116.432
50	5980.068	1957.028	124.9809	567.067	44.07902	-726.911	-270.038
50.00895	5970.862	3005.692	124.495	608.3534	63.46744	-590.618	-368.009
52.01063	6080.636	377.0617	87.97719	454.9015	142.9181	-924.604	-109.174
52.03569	6090	1119.015	89.75152	477.8912	149.7896	-835.889	-178.731
50	5982.229	2179.763	123.896	570.8856	52.53644	-701.153	-292.126
50.03245	5993.584	4517.137	130.8696	610.6551	56.35446	-422.824	-496.983
50	5993.442	4268.566	127.9919	611.2089	57.59764	-452.543	-477.916
50	5973.676	2735.294	125.5732	599.4815	62.16501	-625.863	-344.241
50.00933	5972.474	2634.365	126.2755	571.5126	53.97072	-643.579	-327.537
50	5973.518	2602.614	125.5028	599.7022	62.75163	-641.236	-333.342
50	5971.091	2626.89	126.7504	599.9022	58.59264	-637.205	-333.634
50	5979.916	3282.369	122.91	606.2902	68.08569	-563.148	-394.498
50	5977.007	2954.464	126.4061	621.0771	56.94206	-596.191	-366.267
50	5982.214	3320.579	122.0783	612.1724	60.67896	-557.941	-397.665
50	5971.84	3056.502	125.8661	609.8438	64.10101	-585.077	-373.076
50.02035	5983.387	2860.594	121.6796	606.5149	63.97343	-613.014	-358.747
50.00377	5993.016	4342.596	127.1512	615.669	64.80502	-442.705	-486.644
51.95819	6078.601	560.4559	89.65425	448.2247	144.4201	-905.674	-124.394
50	5969.729	3386.565	125.9535	615.5847	69.63277	-544.407	-402.576

50	5979.06	3187.904	122.6193	608.577	67.77333	-573.206	-386.82
50	5979.862	2050.906	124.8061	569.6284	44.02049	-715.222	-278.324
50.03202	5993.491	4548.456	130.6247	610.7799	56.20499	-419.098	-499.595
51.97184	6073.27	472.7157	87.80903	443.7461	154.7683	-914.594	-117.216
50	5984.743	4055.274	127.653	600.3053	59.60877	-476.439	-456.323
50.01237	5978.784	1963.56	124.0774	572.436	48.21227	-723.988	-272.088
50	5976.309	2129.922	122.6121	572.4969	48.87506	-703.835	-286.028
50	5979.974	2068.33	124.8499	569.6343	46.32399	-713.314	-280.377
Jumlah						22	78

Lampiran 8

Tabel Hasil 4.8 Tabel Hasil 2 Fungsi Objective NSGA-II Wira dengan 100 Populasi 75 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif
50	6087.748	30	208.2353	12	147.0661	-1143.37	-42.78
53.80028	6013.625	5044.395	259.2986	900	273.2132	-196.133	-532.137
52.74667	5988.833	3632.236	253.1831	873.9663	248.4184	-387.228	-429.074
50.00024	6066.622	3884.439	220.4311	342.3001	138.3025	-602.061	-421.344
52.86923	5999.334	3871.662	257.0233	873.6628	256.074	-360.628	-449.444
53.36616	6020.788	4508.987	253.5965	850.693	269.0027	-286.721	-491.991
50.11135	6090	754.6914	210.6505	143.639	136.2065	-1023.26	-123.765
53.39654	6013.681	4781.871	255.9869	896.6533	263.1551	-239.289	-519.534
53.33815	6007.488	4218.29	257.1383	884.8265	247.444	-306.579	-466.364
53.27589	6007.022	4173.104	257.1476	862.5296	246.4277	-319.057	-459.635
50	6090	313.4411	207.9688	12	147.7258	-1111.29	-67.1546
50	6090	2743.927	217.3528	197.029	159.9537	-782.195	-308.82
50	6089.236	1684.333	214.8788	194.0864	150.2319	-905.812	-217.557
53.19041	6020.691	4618.167	253.0514	896.7891	272.1212	-267.776	-516.391
50.00004	6089.262	940.1949	215.6234	12.47206	141.1732	-1037.76	-116.889
50	6089.002	2812.336	218.3642	281.932	163.6683	-752.713	-332.079
50	6090	155.239	203.138	12	149.0681	-1129.63	-54.7898
50	6073.789	3017.445	219.4907	378.111	146.1052	-697.599	-360.1
50	6077.664	2894.429	217.3124	300.9415	146.3606	-732.858	-335.666
50	6065.951	3649.188	219.9488	335.7176	138.6794	-630.865	-400.405
50.00002	6089.895	396.8554	203.1229	12	146.6715	-1101.31	-74.2999
50.0082	6089.153	1421.977	209.9191	128.125	146.1864	-952.328	-181.72
50	6090	2811.177	218.8927	208.8025	156.6019	-771.345	-315.786
50.00001	6090	2383.351	220.0704	236.677	157.2714	-814.351	-285.759
53.24982	6006.838	4315.265	258.8227	896.7069	257.7245	-295.062	-481.71

53.2903	6022.11	4565.148	256.5279	869.5704	271.3289	-278.474	-503.341
53.20081	6004.611	4210.965	255.2169	804.9096	243.294	-330.084	-452.326
50.02268	6089.997	1054.021	210.402	159.793	136.4008	-986.961	-154.673
50	6089.744	2619.999	216.4063	217.9381	153.498	-791.068	-301.091
50	6076.018	2937.052	220.67	367.9787	143.0798	-710.416	-351.058
50.00006	6088.356	1347.238	211.3683	111.912	142.7375	-964.934	-171.287
50	6090	1066.529	212.797	41.25896	145.4387	-1016.2	-134.709
50	6086.619	2166.84	220.0643	233.8576	152.0058	-838.675	-264.991
50	6086.962	1950.777	213.8624	201.49	150.8442	-871.877	-240.928
50	6087.699	2124.152	218.1393	221.1974	151.9642	-847.223	-259.354
53.26295	6011.608	4276.113	257.2096	887.2431	255.372	-303.537	-476.927
50	6089.804	878.8862	215.2237	17.70326	131.2695	-1043.5	-110.527
50.0003	6089.871	2098.352	215.9121	178.6612	154.797	-861.819	-250.162
52.90814	5988.781	3758.288	254.1078	896.758	243.9424	-361.839	-438.068
50	6066.931	3600.64	220.1424	321.3109	139.3435	-640.584	-393.87
50	6090	2276.27	219.9505	238.0685	159.5606	-826.568	-277.698
53.19494	6019.25	4590.698	255.0824	886.0529	278.0818	-273.171	-512.745
50.02308	6089.994	1129.67	210.5308	159.8341	136.0732	-978.105	-160.878
50	6090	1885.497	217.5004	198.6234	152.994	-881.719	-235.848
50	6067.746	3709.151	220.0562	343.8808	133.3751	-622.433	-406.147
50.0003	6084.533	1970.748	212.7325	211.2059	152.6429	-866.079	-244.482
50.00023	6066.44	3798.389	220.0969	342.077	137.6569	-612.044	-413.955
50	6090	2313.491	220.0182	244.4498	158.9557	-820.613	-281.918
50	6089.056	2646.311	215.3752	162.5039	152.8357	-801.502	-291.962
50.00336	6090	1798.502	218.6733	181.1844	151.606	-896.12	-224.545
50	6090	1429.313	217.6537	203.3745	153.0244	-933.759	-198.775
50	6090	2609.849	214.7823	205.1777	149.7138	-795.373	-296.973
50.09328	6064.505	3402.678	221.6371	386.5334	147.2798	-643.839	-388.847
53.19309	6004.441	4212.84	255.2149	806.0777	243.9453	-329.753	-453.068
50	6078.986	2757.335	218.0909	285.9791	153.8099	-753.44	-323.364
50	6090	196.3864	202.9824	21.57213	149.2143	-1122.44	-60.1853
50.01146	6089.507	1581.609	215.196	198.0585	155.2941	-916.757	-210.749
50.00837	6090	1040.905	209.1691	107.7851	142.946	-1002.1	-145.368
50.07262	6065.011	3299.382	221.2488	348.8109	142.2697	-665.98	-372.227
53.688	6013.794	4872.885	256.5362	893.5679	274.2444	-221.135	-520.503
53.78361	6014.617	4990.438	259.8433	882.6944	273.1353	-207.7	-524.854
50.00161	6089.706	1514.191	217.8435	194.1442	153.5577	-926.013	-203.995
50	6090	2131.9	214.0576	275.0146	161.7395	-834.051	-274.19
53.36259	6021.222	4521.303	255.3105	899.4367	277.3289	-273.729	-504.891
50	6072.726	2846.948	218.9407	346.5438	143.9883	-724.833	-338.839
50	6086.648	436.8263	197.6762	12	153.9766	-1095.31	-79.1906

50	6090	2202.463	213.7994	230.7907	159.6743	-836.797	-270.735
50	6089.777	1568.76	218.9145	204.8175	153.5718	-917.093	-210.639
50.00025	6067	3829.788	220.1543	348.2768	139.5911	-607.139	-418.429
53.63238	6012.24	4868.304	256.0027	900	273.5771	-221.017	-522.575
50.03161	6066.005	3302.332	221.1042	332.8101	142.8465	-671.311	-370.91
52.7371	5987.844	3637.087	255.0932	900	248.2536	-380.07	-434.492
50	6089.784	846.9608	215.1543	12	131.0402	-1048.63	-106.671
50.04613	6090	785.2236	216.3517	19.22728	142.6649	-1053.17	-104.428
52.90798	5988.882	3735.288	254.1583	896.7924	242.5036	-364.517	-435.824
50	6090	1448.432	210.5713	146.6329	138.0101	-944.977	-185.975
50	6073.22	2896.233	218.9776	344.829	143.6196	-719.714	-342.631
50	6090	2073.736	219.1728	214.302	155.0114	-855.961	-255.007
50	6087.77	1472.838	215.9438	172.5775	146.0608	-935.125	-194.115
50	6067.049	3239.396	219.8123	339.3334	138.7591	-678.244	-367.287
50	6073.519	2922.012	220.1648	359.2683	146.1072	-713.349	-348.245
50	6065.845	3371.463	220.1456	371.8923	138.7211	-654.184	-384.461
50	6066.563	3713.899	220.137	334.5237	138.2454	-623.871	-405.585
50.00747	6090	1108.333	209.3803	106.2914	142.5218	-994.628	-150.588
53.76228	6014.641	5055.19	258.8919	876.9163	278.3327	-202.382	-531.135
50.00001	6088.506	754.0295	211.5301	12	139.0965	-1059.07	-100.983
50.06944	6090	1311.935	210.7327	112.1178	141.1006	-967.556	-166.37
50	6088.581	2805.952	216.8015	233.6674	155.46	-765.016	-319.893
50.0035	6086.717	645.3213	200.4026	25.39145	152.9228	-1067.61	-98.6198
52.76565	5989.242	3658.876	255.4451	898.8423	246.738	-377.529	-435.156
50	6089.777	1480.033	217.8741	205.5733	153.5786	-927.221	-203.502
53.31921	6011.257	4281.723	258.1672	899.0657	256.0986	-298.143	-478.069
50.00768	6090	1091.691	209.3821	107.4993	142.3945	-996.257	-149.404
50	6090	157.6147	203.5067	21.684	148.9471	-1126.94	-56.8575
53.68512	6009.57	5000.436	259.2259	884.3455	272.7583	-206.855	-527.678
50	6066.834	3378.328	219.8576	339.1109	138.5992	-661.994	-378.722
50	6090	2153.551	216.2691	187.7256	152.2224	-853.104	-255.936
50	6089.648	1833.438	217.3806	191.6839	151.4522	-889.32	-229.66
50	6090	1814.485	218.5451	179.3296	151.5938	-894.816	-225.617
50	6089.047	2654.98	215.3731	162.333	152.8573	-800.531	-292.654
Jumlah						25	75

Lampiran 9

Tabel Hasil 4.9 Tabel Hasil 2 Fungsi Objective NSGA-II Wira dengan 100 Populasi 100 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif
51.0434	5529.484	30.08317	13.73426	619.6376	173.13	-712.617	-19.4125
50.00705	5465.503	5200	216.4456	615.1457	76.39881	-117.482	-419.982
51.15222	5533.604	1880.896	18.51246	773.1099	163.889	-456.693	-199.319
50	5465.937	3259.591	272.1694	391.6149	61.04732	-401.49	-204.485
50.94702	5545.06	960.3553	9.064155	747.5208	166.5169	-581.38	-128.111
50.98189	5538.467	431.5335	18.9659	763.5858	187.6915	-636.192	-88.8646
50.02975	5468.19	3745.267	233.8675	361.5474	69.51612	-351.435	-244.562
51.11835	5533.937	289.1249	16.45109	768.6437	189.4543	-645.46	-73.4748
50.01872	5468.159	5112.925	213.032	608.8817	77.97068	-129.946	-412.521
50.97949	5528.146	849.4593	9.15977	734.8426	167.0406	-589.204	-111.253
51.08909	5533.686	1701.092	12.91268	774.6784	165.6919	-479.081	-187.574
50.03023	5472.377	4064.258	212.2726	503.6894	70.36465	-279.792	-302.977
50.97356	5542.713	897.9373	3.168	748.0973	161.571	-586.343	-120.995
50.00709	5465.215	4083.091	245.4598	325.9849	73.40947	-320.819	-265.352
50.00546	5471.328	3977.971	215.9308	501.6339	72.03788	-290.839	-295.909
51.09149	5540.173	178.9508	18.5804	767.458	194.0391	-662.342	-67.3543
51.04392	5536.995	1095.955	3.168	746.7034	156.4086	-558.829	-132.387
50.07781	5474.033	4423.909	222.3453	511.9291	69.59611	-235.373	-332.383
51.11732	5535.692	83.51391	8.363209	670.6009	204.1423	-694.97	-41.6823
50.01456	5460.779	3752.42	249.3185	335.2593	61.86397	-354.676	-235.045
50.01333	5463.986	3583.508	273.0728	365.368	60.07786	-369.015	-225.008
50.92118	5521.038	1619.182	14.10979	743.297	186.4824	-496.772	-181.425
51.04832	5529.949	106.346	27.94532	618.2555	175.8487	-704.684	-24.7188
50.94699	5521.793	1559.947	11.43943	737.5203	189.8637	-504.704	-175.86
50.93339	5526.324	756.7852	3.168	735.5241	169.2425	-600.313	-105.743
51.09413	5535.641	585.1635	20.35809	788.7666	196.9778	-607.738	-104.812
50.93287	5532.989	600.621	3.168	750.5919	174.2464	-617.836	-98.6758
50.0387	5464	3308.204	245.9512	414.1676	62.48083	-387.354	-214.381
50	5464.508	4156.633	237.0023	533.1016	57.91707	-259.568	-309.912
50	5468.326	3840.373	243.804	363.0716	71.13175	-341.295	-253.129
51.12328	5533.939	264.2911	16.50512	768.6472	189.3369	-648.207	-71.2237
51.1197	5535.889	30	8.403161	671.2928	205.4285	-701.099	-37.6567
51.09291	5543.429	30	11.2627	745.056	194.2068	-686.45	-52.0067
51.01784	5528.398	1474.056	13.45994	760.2352	157.586	-508.774	-164.5
51.10603	5537.653	453.2203	16.68436	770.6248	190.6612	-627.848	-89.1226
51.01804	5535.159	478.2804	27.70228	782.6176	201.6733	-624.22	-97.2768

50.03885	5471.234	5160.077	229.998	566.3963	73.25137	-136.203	-405.241
50.01199	5483.033	4253.124	237.9284	520.228	66.76533	-259.523	-321.77
50	5464.197	5195.31	214.3917	612.3966	70.81405	-118.107	-417.735
50.0177	5468.076	3505.256	263.9085	363.9536	63.31814	-379.941	-220.822
50.03282	5474.947	4490.002	216.964	509.8043	70.76534	-229.799	-339.875
51.15143	5533.414	296.07	3.168	682.6084	206.8988	-665.079	-61.4128
50.04596	5468.9	3935.099	222.613	502.8784	75.84661	-293.612	-291.047
50	5468.393	3917.128	243.2595	376.6443	71.38667	-328.966	-262.375
50.04394	5476.702	4303.33	237.5006	544.2674	70.97857	-244.08	-329.316
50.02488	5471.837	5093.109	207.0157	567.0913	63.59924	-143.435	-400.255
50.04315	5476.181	4279.768	239.3343	539.3395	71.13912	-247.925	-326.117
51.08197	5537.91	99.45972	19.39724	759.8869	196.4324	-672.921	-59.4524
50	5473.654	4730.3	236.3582	540.6105	64.76785	-194.933	-363.283
50	5472.687	3811.681	221.0468	500.0279	73.95034	-311.628	-282.2
50.01691	5464.595	3696.288	212.6699	531.2745	57.2061	-312.427	-272.955
50.03383	5471.008	4981.948	209.5213	573.1521	61.07779	-154.26	-390.847
50	5462.542	3201.62	242.2337	427.617	64.62517	-396.903	-209.893
50.03041	5474.013	4642.481	217.9532	553.8195	67.84477	-200.609	-360.394
50.00653	5467.025	4419.553	212.5422	578.0472	67.5564	-218.058	-346.101
50	5476.194	4255.643	238.6593	525.588	71.7327	-255.475	-322.87
50.90406	5520.297	1378.84	6.269548	752.499	177.3167	-522.137	-162.058
50	5467.469	3595.372	268.8231	381.4693	59.34875	-365.348	-230.73
50.00739	5470.732	3618.977	273.7822	380.766	62.51259	-364.233	-233.446
51.00699	5517.12	1449.135	3.449946	751.3598	170.5866	-509.438	-162.405
50.00686	5466.014	3470.068	260.3775	372.1447	55.86522	-381.064	-217.828
50	5469.826	3851.377	217.742	495.3652	70.01527	-306.681	-283.207
50.02568	5471.667	5037.777	205.9209	567.6906	63.58824	-149.606	-395.804
51.04342	5529.508	71.62413	13.73938	620.3508	173.1177	-707.602	-23.0186
50	5466.432	4071.138	179.8827	522.2182	51.84229	-270.99	-305.285
51.21818	5533.357	1831.989	7.33726	753.7604	164.6255	-464.802	-190.634
50.98599	5521.536	1175.414	12.41287	732.6592	187.6124	-549.457	-140.949
50	5471.718	4929.565	207.9812	574.6853	61.21166	-161.264	-388.168
50	5468.04	4331.074	211.2405	596.1893	64.45546	-224.335	-342.161
50.91293	5520.439	1323.227	6.754901	741.979	175.8462	-531.018	-154.673
50.99755	5521.908	1300.029	6.792677	736.8572	188.4575	-533.524	-152.693
50.0001	5467.388	4922.658	221.6011	551.7382	67.52009	-166.591	-382.133
50.99441	5538.878	1170.173	3.168	740.4741	158.7372	-554.106	-139.864
50	5466.495	4442.868	215.7068	597.8772	66.65779	-210.423	-351.532
50.88364	5519.583	1244.544	5.131424	746.4412	177.3073	-539.584	-150.2
50	5469.82	3884.442	217.6359	503.5359	70.06519	-300.777	-287.618
50.0004	5466.891	4108.371	185.9724	518.5864	59.27849	-268.192	-309.014

50.02753	5474.526	5107.312	212.5743	578.4572	63.30607	-140.197	-403.676
50.01712	5464.48	3697.038	212.7067	531.5044	57.32368	-312.231	-273.054
50	5474.212	4762.521	244.9465	549.01	65.02999	-189.611	-366.995
50	5468.226	3847.19	241.9829	366.3049	70.90077	-339.579	-254.444
51.15099	5533.596	1853.428	18.83113	767.113	163.678	-461.434	-195.781
50.00669	5467.397	4508.121	213.5121	590.1296	70.07824	-204.978	-356.52
50.0346	5471.251	4947.74	209.2324	560.2492	61.01346	-161.547	-385.466
51.14811	5533.795	1778.079	9.358233	762.3144	160.0056	-471.159	-188.707
50.00676	5466.921	4424.589	212.1327	586.8545	67.4431	-215.2	-348.262
51.05032	5537.005	1127.692	3.652174	746.5112	155.6918	-554.979	-134.576
50	5463.554	3967.88	229.5547	331.0022	66.28897	-331.732	-256.362
50.03837	5469.006	4818.029	229.668	559.2443	65.27996	-176.668	-372.805
50	5469.799	3881.462	218.0856	500.4452	69.9156	-301.898	-286.664
50.00816	5467.027	4852.437	229.1184	560.6662	70.87796	-172.516	-377.823
50	5470.396	4892.829	227.2884	559.3445	65.14116	-169.586	-380.76
51.09163	5543.915	68.19238	11.24675	744.8816	194.4644	-682.295	-55.3802
50.03379	5471.307	5023.978	209.2564	570.7198	61.2475	-150.092	-394.008
50.00045	5473.681	4785.396	236.6272	543.5404	63.84776	-187.749	-368.197
50	5474.132	4836.026	245.3956	539.9276	62.72901	-183.21	-370.663
50.01302	5464.227	3904.557	252.0209	365.4407	75.46256	-331.465	-257.798
50.88982	5519.176	1330.417	3.686488	754.5912	176.4435	-527.09	-158.628
50.00781	5468.473	4846.146	212.1484	557.6041	56.40539	-173.604	-375.137
51.04547	5540.367	35.85008	12.99266	766.8422	195.9782	-680.534	-57.7789
Jumlah						37	63

Lampiran 10 **Tabel Hasil 4.10** Tabel Hasil 5 Fungsi Objective NSGA-II Wira dengan 100 Populasi 100 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif	F(3) Leadership	F(4) Egois 1	F(4) Egois 2
56.7424	6065.444	30	100.4997	107.2676	166.6073	-904.674	130.3556	-592.195	-1084.24	-1131.78
50.07871	5894.931	5200	3.168	842.3615	123.6225	-239.078	-603.766	-440.548	-1654.68	-2189.39
54.79307	6073.753	309.5002	130.6266	119.4725	178.0602	-932.481	44.21519	-580.75	-1029.32	-961.443
54.70395	6056.27	30.76859	106.7972	19.61563	176.9509	-984.23	87.00564	-584.588	-1078.28	-1057.18
53.59697	6086.625	4962.286	60.48528	455.6215	99.21757	-342.113	-437.151	-310.032	-1376.44	-1607.26
57.30453	6090	30	130.2644	283.4452	166.4771	-855.394	108.8541	-633.789	-1028.06	-1051.97
50.08725	5877.433	5056.819	3.168	900	118.5063	-233.364	-597.452	-471.282	-1661.63	-2228.78
50.0035	5805.442	4831.193	3.700607	900	118.7682	-231.029	-563.177	-476.749	-1631.53	-2198.17
56.81583	6089.998	39.65524	3.168	900	7.781637	-705.263	-2.95802	-962.119	-1615.62	-2478.46
56.81023	6090	187.4632	9.046624	849.0489	15.81619	-701.389	-6.67409	-924.821	-1583.4	-2392.45
57.49922	6090	250.4944	3.168	466.2471	205.7481	-775.201	37.23255	-720.211	-1493.52	-1970.63
59.51634	6064.804	2107.501	4.753764	491.6601	215.8624	-481.155	-58.1581	-502.261	-1443.74	-1852.04
53.78227	5864.455	59.46752	79.42124	418.3517	150.4731	-823.933	29.0522	-703.015	-1207.73	-1453.55
54.07965	6054.737	3320.342	77.58324	513.797	106.2876	-491.636	-289.563	-467.5	-1311.05	-1554.12
54.16653	5894.182	529.7033	96.91534	390.5429	139.7121	-777.603	3.963283	-643.23	-1147.16	-1320.65
50	5555.694	30	64.13564	879.2569	164.3849	-691.77	-101.958	-842.146	-1320.9	-1771.06
50	5650.761	316.0052	76.49374	900	195.0304	-695.846	-160.132	-827.833	-1324.5	-1709.86
50	5795.115	4056.964	3.168	900	97.81847	-316.27	-490.997	-558.892	-1629.07	-2243.39
59.31123	6090	1621.245	3.168	493.3583	209.7868	-554.277	-29.0684	-563.647	-1464.06	-1902.73
57.25051	6089.595	341.8427	3.168	874.9642	12.43035	-663.206	-11.1538	-915.384	-1600.5	-2438.15
50	5576.787	728.9941	83.16498	858.0958	204.8528	-626.635	-169.462	-747.462	-1260.97	-1576.87
54.19215	5909.238	393.8623	94.32417	381.3998	136.1553	-801.288	14.74676	-659.627	-1159.28	-1346.93
53.96672	6082.631	3625.062	64.72413	525.9035	108.4297	-468.177	-329.56	-455.447	-1370.82	-1651.82

50	5655.775	30	73.61108	900	195.3137	-731.3	-137.911	-859.034	-1336.2	-1740.83
56.68072	6057.499	316.2409	94.55247	187.8691	171.4943	-849.501	88.7013	-590.722	-1122.14	-1208
50.07102	5562.567	678.0963	73.79243	897.2105	207.0863	-614.264	-168.854	-764.912	-1294.64	-1656.36
54.49462	6066.628	30	121.8843	132.429	178.3245	-967.433	56.80297	-619.997	-1064.7	-1039.82
54.12767	5890.256	89.9565	85.77472	363.4584	145.5264	-834.965	43.27372	-681.41	-1176.67	-1384.52
50	5799.285	3919.394	3.419961	894.7871	103.7641	-335.637	-480.994	-569.846	-1628.67	-2238.96
50.38888	5583.844	1387.338	77.27638	893.0419	235.5535	-533.31	-229.675	-681.919	-1284.5	-1582.53
57.07349	6083.859	394.4746	4.072744	843.2025	27.59868	-668.367	-16.7657	-895.246	-1590.7	-2394.7
50	5564.734	149.8504	62.50327	900	170.9494	-676.684	-120.158	-837.964	-1335.3	-1789.73
50	5886.013	5141.772	3.173268	803.3572	119.2664	-253.974	-590.155	-433.77	-1642.77	-2166.76
50	5552.534	1299.616	95.01557	897.4888	210.8384	-540.299	-219.141	-689.548	-1221.21	-1487
53.62709	6085.805	4621.922	59.95941	498.4943	105.2502	-370.028	-417.814	-355.954	-1387.97	-1643.65
50.51457	5516.956	1832.159	79.0725	859.5427	232.3698	-456.988	-238.576	-607.826	-1239.75	-1500.62
50.17926	5722.942	3570.665	15.53736	784.1311	129.21	-366.778	-410.488	-531.518	-1525.22	-2028.83
57.70983	6036.705	150.7597	3.168	562.3201	215.8632	-733.743	43.44219	-742.355	-1491.12	-1997.03
50	5656.89	78.6311	75.92307	900	192.6004	-726.096	-141.333	-854.198	-1328.96	-1727.07
50	5817.646	4651.074	4.311399	873.8625	122.3682	-264.112	-546.935	-487.647	-1628.11	-2185.25
59.2132	6080.927	1780.09	4.152273	485.5309	216.225	-536.958	-42.9238	-541.717	-1457	-1876.64
59.29972	6074.329	1526.271	8.336043	484.3686	224.244	-561.813	-18.8812	-559	-1438.28	-1843.39
54.16376	6066.185	3466.037	73.17347	530.0343	107.2165	-472.901	-305.963	-461.839	-1332.9	-1592.6
53.88594	6085.275	4380.76	57.0033	533.2884	111.8544	-381.592	-398.725	-386.93	-1401.12	-1682.54
50.17946	5723.846	3285.034	14.21441	792.9675	129.0184	-398.228	-388.757	-563.975	-1532.21	-2053.7
55.95701	6089.792	3398.489	46.76603	598.1619	92.2856	-418.804	-264.972	-488.486	-1411.93	-1807.19
50.00153	5788.422	4402.51	3.506243	899.9974	100.0687	-273.097	-518.601	-521.721	-1625.12	-2223.98
57.25074	6090	327.4082	3.877092	841.9026	21.59644	-673.654	-5.65375	-902.495	-1590.04	-2402.9
53.87191	6085.278	4360.186	51.97112	528.2712	107.0089	-385.34	-395.72	-391.657	-1416.93	-1718.56

53.56627	6086.624	4951.431	60.49547	469.2843	99.56441	-340.896	-439.986	-316.011	-1380.44	-1616.93
50.12064	5560.082	1109.503	78.2466	891.8746	202.2853	-562.69	-199.981	-718	-1276.43	-1611.3
50.32722	5736.882	3626.518	9.864114	857.9686	130.761	-343.267	-429.911	-554.131	-1565.43	-2116.47
57.70552	6033.663	153.9717	3.168	563.9677	219.5102	-731.889	42.56463	-740.671	-1490.35	-1992.94
50	5877.456	5200	5.3957	801.6506	115.9702	-243.864	-591.481	-425.226	-1631.37	-2147.89
53.9318	5868.593	30	89.70457	391.8722	145.5029	-831.475	42.52591	-692.442	-1165.56	-1372.38
55.94496	6066.477	3435.505	63.36067	618.6132	106.7455	-400.666	-268.635	-472.836	-1352.25	-1683.08
56.88713	6089.998	30	4.432343	875.634	19.36181	-710.77	2.09033	-949.4	-1603.89	-2441.1
50.15251	5730.243	3677.902	13.17117	853.6616	129.2577	-340.774	-436.207	-547.365	-1554.06	-2091.52
50.2476	5891.001	4984.703	9.445534	793.6858	123.7038	-269.807	-569.433	-439.171	-1616.57	-2116.84
53.92852	5869.347	66.85333	77.71507	401.972	148.4051	-824.773	35.22363	-697.388	-1208.43	-1455.35
53.833	6090	4275.059	56.71222	542.6507	111.8224	-395.208	-394.579	-402.752	-1407.46	-1697.7
54.89428	6081.223	30	108.3297	108.2708	174.497	-967.228	69.57707	-617.651	-1101.93	-1117.21
56.28343	6089.648	3253.478	60.86749	618.1025	107.5251	-421.86	-249.447	-492.572	-1363.32	-1709.47
54.47106	6059.665	245.2967	100.4981	84.01077	176.8087	-951.346	47.81959	-592.139	-1121.45	-1141.45
53.79519	6050.21	3409.392	68.14326	506.8163	107.5314	-489.291	-304.245	-463.709	-1344.58	-1611.61
56.89736	6089.998	30	3.168	890.819	20.85096	-706.674	-1.13855	-954.445	-1611.7	-2457.96
54.46032	5862.619	597.4606	70.00235	443.5371	140.2536	-733.079	1.596076	-655.097	-1231.09	-1510.94
50.1675	5887.251	4935.344	4.537375	850.067	121.2691	-262.004	-577.943	-466.43	-1647.06	-2190.96
59.39276	6090	1611.589	3.168	467.1771	222.8561	-559.938	-23.8489	-550.306	-1455.88	-1870.9
56.68364	6063.216	110.0079	100.7525	223.0361	171.9514	-867.381	98.0181	-621.201	-1112.58	-1199.32
50.4028	5489.531	1879.909	77.59269	856.5039	236.212	-443.724	-239.553	-596.318	-1234.73	-1491.55
50	5556.936	992.3018	76.16007	878.4513	200.822	-581.857	-190.206	-727.8	-1281.14	-1622.47
50.06844	5794.673	4197.046	9.727837	900	100.5866	-297.993	-500.543	-539.544	-1605.65	-2190.63
54.76053	6063.787	30	120.6704	55.45733	178.6267	-977.437	80.6881	-590.147	-1043	-989.558
53.96276	6073.22	4115.701	57.05294	532.1774	112.617	-405.292	-371.282	-408.621	-1394.12	-1681.98

50.02631	5556.195	1201.288	94.73436	900	211.3332	-551.947	-211.727	-700.714	-1223.77	-1494.63
54.00595	5905.711	175.2228	95.12219	370.4831	140.0447	-833.738	29.55324	-678.068	-1156.14	-1339.69
54.06172	6089.487	3951.7	55.71058	506.7978	118.5287	-435.003	-355.249	-417.844	-1397.05	-1679.72
50.00139	5652.049	184.6084	77.36351	899.5864	193.5929	-711.777	-148.934	-841.18	-1322.01	-1710.22
54.8702	6054.393	339.918	122.4656	129.2067	199.9398	-916.252	40.60158	-571.743	-1049.34	-988.222
56.82876	6088.71	3109.491	51.18403	591.2987	96.37058	-427.898	-213.673	-498.181	-1377.6	-1760.73
56.85082	6064.164	270.4337	101.1395	209.8317	172.0716	-847.378	92.07704	-598.252	-1105.05	-1179.6
55.95771	6090	3566.725	49.00591	601.4587	95.04603	-398.588	-280.148	-470.763	-1405.36	-1786.18
53.7092	6090	4757.798	57.44912	493.9885	109.6515	-354.721	-428.173	-340.47	-1395.32	-1648.9
56.36954	6069.626	30	96.49562	68.29842	162.9177	-927.157	126.4403	-589.076	-1097.04	-1145.73
53.46309	6086.21	4791.774	60.13006	466.7692	99.505	-363.054	-429.191	-332.785	-1382.92	-1624.92
53.52064	6077.048	4908.425	59.71692	466.7154	99.76719	-343.756	-434.991	-318.067	-1379.31	-1617.76
50.03771	5615.723	2900.072	4.179206	895.0923	124.8874	-374.536	-354.295	-618.003	-1548.97	-2154.42
50.04605	5721.042	3620.885	9.565405	788.7469	127.9997	-362.699	-419.412	-532.948	-1548.15	-2072.97
50.0375	5616.098	2961.687	5.415271	895.1343	120.4133	-367.398	-358.296	-612.826	-1545.02	-2148.82
54.06253	6090	4105.675	57.70584	518.0226	117.7233	-414.471	-370.028	-405.575	-1393.4	-1669.81
54.71087	6058.358	31.52327	107.2455	58.2024	176.9329	-975.207	78.96058	-597.575	-1087.16	-1080.81
54.8478	6060.66	118.4984	98.87189	12	178.2906	-973.266	83.30853	-575.732	-1101.75	-1099.99
50	5551.069	739.9899	70.33259	894.9321	203.3104	-604.519	-172.214	-759.142	-1302.24	-1675.3
50.00457	5806.28	4784.452	3.466755	899.9404	120.2741	-236.871	-559.847	-481.212	-1632.62	-2200.14
50	5792.393	4071.684	3.168	900	105.9957	-313.646	-493.587	-554.014	-1627.94	-2233.36
56.82582	6062.36	113.4735	103.8284	193.1313	170.1855	-869.861	108.9386	-607.59	-1091.66	-1157.26
50	5547.85	1018.25	72.12167	900	201.4508	-569.392	-194.967	-731.926	-1296.2	-1657.72
56.25273	6089.647	3242.623	60.87769	631.7654	107.8719	-420.643	-252.282	-498.551	-1367.31	-1719.14
53.80832	6090	4723.208	57.48866	485.8804	110.3669	-357.834	-420.87	-339.481	-1391.17	-1641.69
Jumlah						20	73	7	-	-

Lampiran 11

Tabel Hasil 4.11 Tabel Hasil 2 Fungsi Objective NSGA-II Kirna dengan 100 Populasi 20 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif
53.62336	6088.775	1749.355	285.2575	29.89808	19.64155	-1089.84	40.7491
53.52038	6070.325	1795.107	285.6	14	31.2121	-1091.15	40.86995
53.43734	6086.326	1894.975	285.5118	88.84789	15.12226	-1004.39	20.36352
53.6089	6090	1850.574	283.8267	80.87276	21.67942	-1019.43	25.77763
53.48638	6090	1743.375	283.2541	69.72898	20.39371	-1054.43	28.1992
53.59783	6090	1741.529	285.5808	33.14271	22.62681	-1089.23	38.5201
53.5933	6084.702	1747.828	285.3405	23.51389	11.51364	-1095.29	44.36767
52.33089	5297.336	2939.771	241.3336	633.5531	190.7447	53.70046	-9.86446
53.61881	6090	1828.86	285.4915	28.11001	18.75197	-1076.31	38.27063
53.43985	6086.387	1894.902	285.5693	88.75773	15.13016	-1004.47	20.44658
53.45687	6086.659	1895.165	285.3755	83.59951	14.98511	-1009.34	21.9343
53.62138	6088.697	1750.161	285.235	30.19102	19.57197	-1089.39	40.64041
53.56938	6081.932	1862.479	284.9395	36.19805	17.88066	-1059.1	36.25298
53.55221	6087.024	1895.177	285.3644	77.95068	15.93853	-1013.25	25.57787
53.44065	6086.382	1894.919	285.5078	90.77003	14.93439	-1002.43	20.11582
53.43721	6086.418	1895.046	285.5097	87.6909	15.32263	-1005.58	20.51187
52.31547	5402.287	2902.682	242.827	692.526	210.2602	60.86725	-52.2028
52.42467	5243.016	2848.526	241.9249	624.918	196.6736	51.21681	9.766002
53.42874	6082.644	1901.665	284.8453	104.5252	13.00791	-985.966	18.17561
53.57985	6089.192	1885.45	282.386	119.6906	25.28897	-973.913	15.0997
53.62617	6081.924	1862.446	284.8539	37.7446	18.01714	-1056.43	37.60699
53.55094	6086.609	1895.15	285.4404	83.60984	14.9777	-1007.44	24.77507
53.44846	6069.038	1894.141	283.9284	128.7056	10.77471	-957.174	18.10274
52.43707	5366.736	2861.908	242.5699	721.0768	193.2511	98.79604	-39.5038
53.53618	6081.272	1880.208	284.4615	128.6775	4.631509	-963.239	19.82851
53.55241	6086.207	1895.162	285.3627	84.19506	15.96584	-1006.67	24.53218
53.62607	6082.113	1858.475	284.8768	37.85398	17.88226	-1057.19	37.70494
52.06385	5266.288	2861.974	239.725	649.9384	218.3739	61.68854	-18.2992
52.48828	5278.53	2935.445	237.3358	777.3115	257.8258	206.5261	-47.2699
53.54803	6086.178	1896.213	285.4298	83.2632	15.00362	-1007.45	24.8213
53.62096	6088.648	1751.305	285.2356	30.44703	19.62253	-1088.89	40.53759
53.64706	6090	1916.929	283.7232	72.98469	21.44732	-1013.27	26.3416
53.62059	6088.69	1750.462	285.2287	30.37981	19.60235	-1089.15	40.56236
53.63268	6090	1859.216	280.3758	44.7052	17.06882	-1053.26	34.37846
53.58655	6088.56	1754.731	285.2678	30.43896	19.51929	-1088.86	39.44479
53.62656	6088.885	1803.166	282.9537	14	10.07074	-1094.75	44.56417

53.58264	6090	1919.891	285.2853	42.88487	18.18119	-1044.04	31.33121
53.48914	6090	1846.411	284.7624	50.26625	4.024093	-1052.98	33.23109
53.59391	6090	1775.235	285.5909	31.14675	18.22473	-1084.49	38.85276
53.5511	6086.561	1894.5	285.3816	83.47577	14.81332	-1007.67	24.879
53.53603	6081.319	1880.858	284.5203	128.8116	4.79589	-963.001	19.72458
53.62295	6081.996	1860.029	284.8756	36.31962	17.90701	-1058.43	37.88882
53.58308	6090	1919.63	285.2807	43.3157	18.16192	-1043.65	31.27195
52.40645	5267.035	2864.74	242.8939	640.0191	197.925	59.15124	-0.59925
53.43816	6086.146	1895.084	285.5279	87.90579	15.01122	-1005.22	20.64863
53.63208	6090	1859.254	280.3152	44.37714	17.08615	-1053.59	34.41528
53.53288	6081.216	1880.264	284.4654	126.7554	4.819376	-965.195	20.07621
53.62229	6090	1875.875	284.726	29.17456	11.369	-1065.64	38.50011
53.55149	6082.019	1875.531	284.8662	41.79471	25.16082	-1051.41	32.19251
52.51342	5275.977	2884.549	245.17	634.5027	190.2718	56.09964	3.047727
53.54652	6086.144	1897.653	285.4256	81.51559	14.92348	-1008.92	25.10727
53.63902	6082.319	1864.515	284.8658	39.04292	18.60587	-1054.63	37.40905
52.43536	5321.06	2913.048	243.7079	727.307	199.8625	134.1227	-32.755
53.68588	6090	1824.638	285.571	14.00604	4.006079	-1089.68	47.18325
52.46464	5254.507	2841.993	244.9437	643.4057	197.5128	64.3449	4.640115
53.50052	6071.948	1892.547	284.9428	138.4291	6.859284	-947.893	18.17397
52.47601	5322.071	2883.488	245.8355	665.9469	193.0997	67.31956	-16.55
52.16047	5239.288	2855.173	241.2518	640.6917	194.4648	64.63708	-0.07159
52.41849	5321.037	2894.223	240.7244	697.0666	209.4698	99.67876	-29.3904
53.55537	6086.594	1895.148	285.4405	83.56003	14.9781	-1007.39	24.92174
53.49066	6090	1848.905	284.8064	52.13928	4.022953	-1050.58	32.82288
53.72151	6077.672	1801.099	284.3472	14	23.80688	-1088.85	46.73782
53.62064	6088.633	1750.68	285.236	30.80314	19.4369	-1088.66	40.53066
52.39302	5311.825	2901.845	242.1085	708.2307	208.299	115.6928	-29.9108
52.56963	5275.074	2874.12	243.9745	643.4053	194.2173	64.37139	2.375234
53.48619	6086.319	1894.967	285.5138	88.90696	15.10869	-1003.35	21.82269
53.63452	6090	1859.089	280.2248	46.60232	16.99061	-1051.34	34.06659
53.55117	6086.173	1895.152	285.4374	84.05898	14.99565	-1006.8	24.79582
53.63925	6081.817	1862.803	284.8635	42.24368	18.8635	-1051.56	36.88959
53.52154	6071.609	1890.224	284.8611	128.1764	7.513266	-958.058	20.83631
53.5963	6089.948	1883.967	283.564	107.5345	14.0168	-986.184	20.98889
53.62512	6088.676	1750.297	285.2364	35.88979	19.73391	-1083.58	39.57071
53.6279	6081.924	1863.363	284.8552	37.65875	18.02396	-1056.29	37.64367
53.48954	6089.93	1847.69	284.8043	52.03056	4.005	-1050.92	32.87384
53.56696	6089.728	1897.761	283.6743	23.62502	4.231279	-1067.67	39.10452
53.59718	6090	1883.257	283.5569	103.8337	14.33342	-990.036	21.68105
53.6394	6082	1858.905	284.8592	42.26157	18.83451	-1052.4	36.98212

53.63981	6082.012	1859.081	284.8657	36.24105	17.93162	-1058.36	38.43078
53.62313	6088.797	1759.296	285.2625	30.49145	19.59621	-1087.27	40.29964
53.64074	6082.628	1862.708	284.8438	36.01918	19.08552	-1058.12	37.91875
53.50731	6071.857	1892.333	285.0113	129.3775	6.775881	-956.813	20.2458
53.62713	6081.877	1864.138	284.8715	32.07665	18.00327	-1061.72	38.72965
53.50989	6072.018	1891.505	284.8603	136.3883	7.093663	-949.987	18.81085
53.58877	6089.981	1883.796	283.491	105.1469	14.30062	-988.774	21.15605
53.56786	6090	1899.174	283.8565	14	10.70248	-1077.22	39.23058
53.56806	6089.395	1899.913	283.7014	25.43178	4.005	-1065.27	38.85024
53.50944	6071.022	1892.467	284.9364	135.271	7.023402	-950.506	19.26296
53.56975	6090	1897.414	283.7846	14	10.75402	-1077.53	39.3263
53.51241	6071.971	1892.219	284.8693	134.2042	6.63014	-951.95	19.43582
53.5972	6090	1884.024	283.6224	104.2645	14.2253	-989.451	21.60404
53.50799	6072.257	1891.239	285.2502	136.3108	6.692692	-950.254	18.8583
53.50957	6072.383	1892.058	284.8151	129.6734	7.105697	-956.749	20.02773
53.51227	6072.142	1890.652	284.8615	132.8089	6.672914	-953.733	19.70818
53.68261	6090	1827.197	285.5904	14	4.005	-1089.23	47.003
53.50642	6072.546	1893.284	284.8724	130.9902	6.044184	-955.301	19.8763
53.51206	6072.836	1892.351	284.8821	131.9168	5.928752	-954.567	19.85013
53.62085	6088.38	1758.677	285.2108	30.31886	19.67755	-1087.44	40.36435
53.68479	6090	1825.948	285.5946	14.00484	4.006404	-1089.44	47.10894
53.62074	6088.569	1754.706	285.2337	30.42823	19.54268	-1088.2	40.4623
53.50914	6071.812	1892.458	284.808	132.2663	6.703699	-953.84	19.73217
Jumlah						15	85

Lampiran 12

Tabel Hasil 4.12 Tabel Hasil 2 Fungsi Objective NSGA-II Kirna dengan 100 Populasi 50 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	Health Pahlawan
50.00003	5603.252	1941.238	282.6482	345.1153	121.3228	-608.058	-43.3418
50.00003	5603.843	1931.546	282.7378	343.2887	121.3188	-612.07	-42.7925
50.02853	5610.604	2235.367	285.1201	460.8651	132.5928	-436.205	-80.0785
50	5624.767	2031.25	275.6524	384.0865	122.5159	-559.953	-60.4173
50.00003	5603.135	1935.065	282.2035	346.046	121.198	-608.303	-43.2968
50	5597.58	2265.717	271.9881	500.6316	130.2578	-385.254	-87.115
50.00008	5601.779	1936.011	282.2582	332.9273	121.2386	-620.668	-40.3703
50.03616	5610.518	1969.732	282.3177	334.4066	115.4911	-615.27	-41.3548
55.01685	5935.378	2255.901	272.9986	19.22038	194.3533	-910.126	58.54246
50.02925	5611.777	2245.144	285.0887	457.0487	132.6483	-438.54	-79.93

50	5595.826	2212.345	261.7006	519.7958	124.1462	-375.76	-87.9578
50.0004	5603.113	1936.091	282.2415	358.5629	121.9678	-595.578	-46.0197
50	5596.05	2218.393	261.7076	516.968	124.6114	-377.479	-87.7732
50.00036	5603.126	1924.056	282.4141	341.7201	121.2829	-614.826	-42.0575
50.00003	5603.144	1946.621	282.5001	339.7951	121.4905	-612.257	-42.4873
54.51847	5930.71	2264.278	272.0841	103.0481	193.2018	-832.611	27.94432
50	5602.515	2260.926	270.9244	577.523	130.0137	-311.355	-103.591
50	5624.732	2032.768	275.6198	373.713	122.4739	-570.007	-58.3759
50	5596.446	2208.636	261.9945	515.3839	124.5504	-381.184	-87.19
50.00021	5595.964	2216.421	260.4359	516.0631	124.7333	-378.72	-87.6372
50	5611.766	2232.026	284.7182	454.3656	132.5887	-444.42	-79.846
50	5603.295	1895.006	279.7437	359.6431	121.6346	-602.752	-45.0431
50.02971	5610.005	2210.189	285.0863	451.1214	132.5843	-450.71	-77.106
50.00004	5603.33	1935.225	282.2929	356.6094	120.6121	-597.781	-45.2995
50.00029	5596.086	2217.03	262.0288	517.6097	124.7103	-377.126	-87.8559
50.00003	5603.455	1946.71	282.5677	338.7922	121.5217	-613.373	-42.3701
54.64053	5921.84	2255.937	273.2214	25.6303	194.523	-905.601	49.32764
50.02069	5613.2	2237.439	285.4135	448.5709	131.9985	-449.317	-78.3897
50	5614.338	2190.859	268.81	541.5405	133.6169	-366.302	-98.1517
50.17225	5621.7	1947.715	282.4075	313.5407	122.3772	-642.593	-36.9894
50.00003	5603.522	1941.82	282.6002	342.6697	121.2185	-610.497	-42.9159
50.00584	5651.832	2041.269	274.9242	415.809	125.6435	-537.426	-74.5814
50.02322	5607.484	1951.834	285.6	360.1205	121.4807	-592.285	-46.8546
50	5625.145	2032.728	275.6216	373.0751	122.3807	-570.823	-58.3252
50.02893	5631.049	2041.42	275.6055	392.9205	129.1472	-551.233	-64.998
50.00037	5598.248	2232.482	260.8699	516.9119	124.6183	-375.612	-88.8417
50.00003	5603.135	1934.989	282.2056	346.4539	121.1973	-607.911	-43.3755
50.00003	5603.126	1938.331	282.393	357.0129	121.1525	-596.682	-45.5687
50.00165	5625.966	2235.866	283.5688	467.7879	133.384	-436.107	-86.4667
50	5614.336	2192.19	269.0314	543.4541	133.5574	-364.125	-98.544
50	5611.256	2268.306	272.5362	548.8234	133.6581	-342.309	-101.12
50.03017	5611.765	2244.557	285.0043	457.4933	132.648	-438.188	-79.9759
54.61678	5707.038	30	212.8537	14	131.6432	-1271.34	190.5767
50	5603.216	1938.076	282.3959	357.383	121.1382	-596.401	-45.6536
50.00008	5602.885	2260.388	270.8037	582.503	129.9588	-306.632	-104.654
50.02882	5631.328	2040.984	275.3201	394.4612	129.1545	-549.894	-65.3904
50.00003	5603.752	1938.31	282.6437	344.5695	121.2605	-609.396	-43.2437
50.00006	5624.879	2032.98	275.2218	380.0424	121.9572	-563.68	-59.5792
50	5597.483	2258.151	271.8014	498.5116	130.5634	-388.849	-86.5117
50	5614.391	2189	268.8	538.7056	133.3718	-369.527	-97.4713
50.00036	5629.023	2034.814	275.9092	405.5059	123.5761	-539.608	-66.1344

50	5624.688	2017.329	275.6443	377.8707	122.9074	-568.926	-58.7954
50.00135	5602.381	2262.33	270.8821	577.88	129.9447	-310.633	-103.62
50.02718	5610.737	2233.963	285.1624	460.5081	132.6617	-436.927	-80.0489
50.00003	5602.906	1957.587	282.3809	357.1229	121.8141	-592.64	-46.3551
50	5629.437	2018.365	275.5465	372.4587	119.4508	-576.05	-58.0211
55.2219	5941.432	2257.313	273.3511	14.36914	197.3395	-913.172	63.33676
50	5598.709	2260.898	271.0024	603.3333	131.0436	-283.983	-108.069
50	5624.971	2032.696	275.6803	389.8405	122.4511	-553.994	-61.6477
50.0059	5651.563	2041.036	274.9284	415.1198	125.5184	-538.047	-74.3329
50	5625.036	2031.483	275.6482	384.7757	122.6411	-559.331	-60.6639
50	5611.268	2248.987	272.282	546.8198	133.6634	-348.177	-100.101
50.00612	5652.169	2042.089	274.8988	417.815	125.631	-535.391	-75.0846
50	5614.001	2190.039	268.8354	539.5346	133.6293	-368.333	-97.6401
54.6308	5918.611	2262.316	272.4803	62.82874	196.8731	-866.002	41.5022
50	5611.378	2274.817	272.4694	545.6822	133.9739	-344.203	-100.829
50.07617	5613.28	1967.557	281.034	342.6774	121.197	-607.859	-44.0555
50	5597.345	2256.569	271.7892	500.8876	130.3093	-386.727	-86.8328
50	5611.435	2267.443	272.4541	544.057	133.6669	-347.321	-100.192
50	5611.292	2262.211	272.236	562.0714	133.6216	-330.289	-103.59
50	5617.196	2297.23	274.9963	545.7423	133.7031	-342.122	-102.76
50	5614.439	2191.26	268.6785	538.6006	133.4463	-369.199	-97.5676
50	5624.667	2030.848	275.7839	387.0264	122.6864	-557.056	-61.0014
50.02943	5611.751	2246.415	285.0832	457.1125	132.6407	-438.207	-79.9716
55.36301	5949.672	2254.179	272.8542	14	192.0608	-914.682	67.05485
50	5625.214	2037.449	275.58	386.5582	123.6002	-556.444	-61.5251
50	5603.217	1949.073	282.3056	347.5761	121.5846	-604.015	-44.1857
50.00003	5603.665	1939.259	282.7693	357.4739	121.1859	-596.267	-45.8043
50	5624.605	2031.724	275.2474	373.022	122.3467	-570.845	-58.1683
55.42682	5710.028	30	208.2307	14	131.6933	-1256.31	213.7321
50	5602.471	2261.004	270.9874	577.5772	130.0193	-311.268	-103.59
54.87273	5704.037	141.0641	210.1875	14	131.6899	-1242.71	195.0687
55.20467	5939.509	2256.911	272.2385	14	191.3406	-913.046	64.89473
54.6317	5903.607	2256.223	273.233	69.72575	194.9446	-854.035	44.68093
50.28432	5610.759	1937.83	282.3959	323.2049	120.6525	-628.077	-32.0366
50	5644.927	2162.888	276.4417	406.9451	123.066	-519.189	-74.498
50.03002	5611.754	2239.198	285.2096	448.175	131.8667	-448.567	-77.71
50	5588.57	2221.653	262.5295	531.9515	126.279	-358.769	-89.3848
54.67514	5905.698	2242.719	273.1693	35.41265	192.969	-891.017	53.29543
50.00003	5603.101	1956.988	282.565	359.8689	121.1463	-590.087	-46.7396
55.1824	5713.111	30	212.566	29.05958	131.6548	-1247.49	202.9881
50	5611.288	2283.16	272.4274	545.1102	133.2243	-343.055	-100.773

50	5624.74	2026.839	275.4524	369.5876	122.8714	-575.325	-57.4752
54.60983	5886.163	2244.323	273.3989	59.8099	192.4047	-859.46	51.45676
50.26258	5609.758	1953.275	282.4175	327.3823	120.4986	-620.825	-33.7457
50.00588	5649.35	2030.977	275.8369	418.7748	125.6832	-535.5	-74.1444
55.26362	5706.676	50.37488	205.6038	14	131.6866	-1254.06	208.7776
50.00003	5603.144	1946.754	282.4972	340.0512	121.4899	-611.974	-42.543
50	5624.457	2031.161	275.5847	385.0893	122.4848	-558.837	-60.5345
50.03929	5629.709	2036.782	275.6603	394.3752	129.2168	-549.942	-64.5024
Jumlah						0	100

Lampiran 13

Tabel Hasil 4.13 Tabel Hasil 2 Fungsi Objective NSGA-II Kirna dengan 100 Populasi 75 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif
55.81328	5783.2	141.7193	103.9038	18.63159	287.2157	-1252.94	152.2159
50	6089.796	4800	135.1193	825	266.6099	240.8897	-407.285
53.9356	5841.618	1306.166	110.5317	27.43054	297.3815	-1073.43	38.54789
50.00037	6074.598	646.7646	108.6967	353.0524	183.9906	-1053.55	-150.814
53.9891	5857.942	1303.405	110.9514	14	296.5879	-1093.14	39.09654
55.39672	5786.142	781.927	100.0505	39.11873	289.6742	-1113.95	112.5691
55.35908	5806.926	594.8655	101.2203	14	286.249	-1185.86	118.5138
50.00036	6075.48	808.8187	107.0277	361.5843	174.8121	-1012.79	-155.834
55.78787	5779.671	420.7648	102.9184	14	287.3452	-1200.79	143.844
50	6085.776	980.3447	108.1976	365.4595	178.405	-978.988	-165.772
50	6082.468	2154.513	130.7951	524.8682	283.5164	-585.496	-262.113
55.72898	5786.346	471.2475	103.2555	15.30622	289.406	-1193.39	137.9432
50.00003	6089.914	1388.138	106.299	665.2864	186.2198	-599.425	-242.607
55.36562	5789.53	773.7471	100.4197	16.51463	289.5038	-1140.23	115.6591
50	6089.676	3612.549	119.1254	616.7979	298.5213	-205.018	-335.876
50	6089.999	3040.565	118.4066	609.9543	298.7	-326.384	-315.629
50	6090	4236.883	119.8139	609.1438	290.6322	-87.8203	-353.076
50	6089.788	4702.346	137.7794	817.2469	271.752	213.4789	-403.627
50	6086.801	2306.062	132.2983	533.6886	283.3915	-548.194	-269.853
50.06039	6082.844	645.8218	108.7424	359.9379	184.9415	-1049.1	-152.67
50	6088.512	464.0213	113.2299	594.0207	177.545	-854.901	-194.309
50	6086.25	3297.188	133.9651	531.4872	288.5453	-352.054	-303.548
50	6088.83	4339.038	133.9056	825	273.7527	149.0009	-393.684
50.00049	6089.606	826.8394	107.4544	534.7233	176.9657	-841.975	-195.129
50	6084.541	1226.748	106.0371	635.3152	210.7876	-659.841	-236.463

50.00041	6089.972	3265.746	119.8625	587.347	298.7	-303.96	-318.473
50	6089.979	4139.887	138.0974	779.2145	282.5311	62.69003	-380.167
50	6089.992	3549.988	119.3523	600.756	298.382	-233.706	-330.605
50	6088.134	1014.1	111.5193	606.6713	180.8092	-732.103	-216.094
50	6087.923	939.2981	110.6125	602.2055	175.9758	-751.346	-211.441
50.00012	6089.69	4450.645	137.3995	786.545	280.4199	132.3422	-391.411
50	6088.542	628.4412	112.9511	598.9321	172.6507	-817.032	-199.498
50.0029	6074.592	997.7933	108.5346	383.7998	179.7428	-952.468	-167.468
50.00012	6089.755	4457.545	137.369	767.9619	280.4158	115.1125	-387.942
50	6089.731	1194.332	107.4281	385.2083	180.6389	-918.114	-178.504
50	6088.544	907.4391	112.941	598.9893	170.7577	-761.144	-208.305
50	6090	4049.254	137.3877	740.1577	278.7545	5.572866	-368.392
50	6080.815	1230.978	106.1155	646.1499	211.1844	-646.616	-237.939
50	6086.916	3088.572	129.9674	531.8992	294.7469	-393.68	-298.83
50	6090	1101.019	106.7288	377.3163	178.505	-944.734	-173.371
50	6084.436	1144.7	104.8884	628.1131	211.1787	-683.396	-232.462
50	6089.765	4581.417	137.3053	816.2874	277.2798	188.259	-400.912
50	6088.102	1201.409	110.467	608.6846	179.7883	-692.58	-222.547
50	6089.687	4686.904	137.8529	780.3695	271.4871	173.5585	-395.635
50.0558	6089.989	2815.255	132.8186	554.5426	281.1248	-425.685	-289.472
50	6077.458	2400.922	130.6968	563.9982	283.2822	-494.991	-276.845
50.00045	6089.318	689.2035	107.2029	529.0914	169.7677	-874.891	-187.446
50	6073.796	2122.999	130.8314	550.4245	283.4977	-562.63	-263.997
50.00002	6088.172	1142.438	111.623	604.8353	185.7557	-708.371	-221.324
50.04161	6089.291	4099.967	119.8656	619.0658	290.4347	-104.152	-349.014
50.00147	6090	4029.307	137.929	765.2454	279.1085	26.68562	-372.75
50	6082.394	1119.037	111.8179	599.7768	180.7259	-715.622	-216.731
50	6090	4210.252	135.3545	787.0473	273.413	84.78493	-381.882
50	6078.585	2354.677	131.0056	542.248	283.7997	-526.474	-271.348
50	6088.689	4566.379	119.3158	604.5122	290.7135	-25.9994	-362.868
50.01624	6090	810.2174	111.2111	593.4895	181.7993	-786.526	-206.93
50	6073.973	2501.2	134.2236	570.8515	280.4181	-466.641	-279.63
50	6089.476	3998.497	118.8386	610.4247	298.7	-134.116	-347.487
50	6089.593	3805.131	119.6725	611.3233	297.478	-171.933	-340.855
50	6072.801	2629.91	134.0589	561.6682	283.8496	-449.649	-282.719
50	6089.925	3879.149	134.8371	707.1526	270.1934	-61.2368	-354.032
50.00007	6073.984	2435.022	133.376	572.8654	282.2075	-477.882	-278.375
50	6089.54	3863.221	119.0364	612.4969	296.1135	-159.086	-342.702
50.06551	6090	1139.276	105.0428	622.6716	213.5388	-690.973	-231.235
50	6089.97	3236.093	118.8061	554.4859	298.1876	-342.733	-310.876
50	6089.781	4742.817	137.9994	825	271.9205	229.3228	-406.551

50	6071.901	2289.072	128.0793	573.8046	283.7932	-505.204	-274.043
50	6089.149	1268.828	106.0891	668.588	184.7537	-619.639	-238.726
50	6089.843	4083.871	119.5288	613.1328	290.4382	-114.36	-348.706
50.05693	6090	2792.413	132.5857	552.6608	280.7876	-432.107	-288.232
50	6089.99	3059.281	132.222	525.0958	284.6259	-407.491	-294.374
50	6088.535	3173.221	132.8695	535.5189	285.6854	-373.703	-300.122
50.00034	6089.517	4002.629	136.3119	742.1587	265.0149	-1.29644	-363.533
50	6090	3509.517	119.1324	593.0028	298.2134	-249.55	-327.68
50.00048	6089.382	558.6631	107.3664	526.0711	167.8529	-904.016	-181.981
55.39292	5788.792	756.9598	100.6877	33.14256	289.4319	-1126.11	113.9378
50.07306	6090	3099.202	119.1905	579.0477	298.0634	-344.105	-308.976
50	6072.782	2272.406	129.301	563.8923	288.0705	-518.909	-272.765
50	6090	3733.453	119.0581	612.3149	298.7	-185.457	-339.143
50	6089.977	4097.835	138.3853	773.0637	279.9231	48.16829	-376.816
50	6090	3726.308	119.0676	607.0221	298.4462	-192.175	-337.778
50	6074.016	2114.631	131.085	546.3992	283.4438	-568.423	-262.933
50.00045	6089.316	686.3839	107.2033	529.2119	169.8058	-875.334	-187.385
50.00045	6089.38	607.0493	107.1773	525.9455	170.3923	-894.503	-184.262
50.00148	6089.898	4034.494	137.3931	766.1985	282.4268	28.67229	-374.017
50.0108	6090	4022.731	138.2089	784.3469	279.9765	44.63928	-376.279
50	6089.99	3051.461	132.1221	527.4945	285.2464	-406.665	-294.767
50	6090	3336.374	118.7795	615.7313	298.4759	-261.448	-326.554
50	6072.819	2618.158	134.0731	561.5394	283.8547	-452.136	-282.306
50.00003	6089.909	1378.951	105.9913	662.4764	185.1896	-604.048	-241.488
50	6088.535	717.042	112.7847	609.2301	174.6108	-789.041	-205.045
50.00049	6089.682	3583.57	118.5281	590.7882	295.7318	-236.76	-329
50	6090	1115.157	106.7196	371.0425	178.6673	-948.182	-172.632
50	6089.909	4280.591	133.6126	789.5172	273.4756	101.3886	-384.859
50	6090	1173.2	107.3144	392.4328	184.9977	-915.299	-180.483
50	6090	3963.759	119.3025	607.4087	291.353	-144.184	-343.859
50	6089.122	1254.69	106.0983	674.8618	184.5914	-616.179	-239.459
50	6088.511	715.6964	113.0899	598.6428	174.5678	-799.892	-202.84
50	6088.538	3159.368	132.9766	537.472	285.3122	-374.517	-299.943
50.00119	6089.599	3763.183	135.8096	740.2008	268.3433	-51.2077	-356.084
Jumlah						35	65

Lampiran 14

Tabel Hasil 4.14 Tabel Hasil 2 Fungsi Objective NSGA-II Kirna dengan 100 Populasi 100 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif
51.00127	5774.909	30	251.1621	14	143.4604	-1372.76	65.18493
50.38621	6089.944	4800	136.2471	825	295.9571	248.0443	-403.467
50.03731	5849.894	1967.682	263.7142	315.1565	171.1745	-735.265	-113.645
51.33396	6089.999	1865.723	123.6768	413.3237	298.6856	-731.392	-196.68
50.20846	5791.405	30	265.6924	165.1727	176.3379	-1245.11	-0.51429
50.9815	5773.068	216.7077	251.8389	38.17561	146.1404	-1310.93	53.33493
50.16169	6090	3281.24	136.5508	731.4801	277.9302	-153.446	-336.056
50.00005	5833.579	605.828	280.3501	284.2245	156.8037	-1032.55	-53.8841
50.49181	6090	4018.162	150.8605	811.345	297.0298	79.84871	-370.59
50.07919	5785.972	36.90682	259.3735	192.4168	168.1274	-1216.56	-7.05041
50.4873	6062.349	3653.381	133.1358	633.5672	289.5492	-159.034	-315.58
50.01793	5860.531	30	265.7342	230.2841	157.3199	-1212.3	-31.4592
50.04068	5839.841	391.0203	260.2328	265.8302	183.7382	-1095.82	-52.2508
50.23188	6089.984	3619.049	143.4632	818.1643	298.5713	1.751216	-367.471
51.24527	6090	2003.924	121.1347	456.1018	298.7	-662.705	-212.719
50.04307	5866.919	1484.718	266.3754	194.4561	150.8059	-959.242	-71.8365
50.03034	5843.895	1745.987	274.3345	318.9961	168.886	-773.543	-104.237
50.95126	6089.977	3265.373	70.00518	585.0844	298.7	-286.451	-293.639
50	5829.069	832.5833	272.9797	282.994	158.7338	-986.463	-61.1994
50.1544	6090	3394.895	125.3773	776.3073	295.0925	-86.1335	-354.536
50.90601	6090	3450.351	82.31947	591.5424	298.7	-244.117	-301.433
50.03065	5863.242	2064.563	258.9601	289.8034	157.4655	-746.629	-112.081
50.93566	6090	3452.763	81.87978	584.442	298.7	-250.135	-299.241
50.00749	5857.184	1706.711	271.4843	289.5096	156.7243	-816.629	-98.0331
50.57326	6070.51	3551.795	137.2649	610.4671	292.6756	-204.253	-307.524
51.01414	5780.526	170.78	254.192	25.9398	134.8429	-1334.66	59.63665
51.43634	6078.995	2044.56	123.7356	498.0482	295.3755	-604.213	-212.876
51.1201	6089.952	3151.52	92.83909	543.0209	289.9787	-348.133	-272.131
50.46145	6087.129	4223.263	143.6519	805.2658	298.461	115.4753	-377.386
50.43629	6085.994	4229.478	143.3433	819.8816	296.9022	131.3349	-380.597
50.52672	6090	4024.808	146.3236	825	298.1841	95.58757	-373.181
50	5835.665	794.6969	273.5588	273.3818	157.4698	-1006.39	-59.2778
50.01148	5856.262	1569.385	269.9603	221.3018	153.001	-911.75	-78.5981
50.03572	5846.004	356.1746	255.2456	251.3752	181.6504	-1119.79	-49.7465
51.46001	6088.389	2155.229	125.6395	519.0599	297.3213	-564.572	-222.766
51.00414	5768.079	108.6266	250.5788	18.14878	145.2805	-1350.01	62.9939

50	5844.429	30	249.4282	268.787	177.1982	-1167.5	-42.3318
51.32513	6088.377	2753.987	120.5834	503.6981	298.7	-462.814	-244.484
50.14788	6086.123	3387.887	124.6012	723.8456	296.3314	-138.519	-343.431
51.41481	6089.944	1963.955	118.9639	448.7149	298.5921	-674.634	-204.961
51.20399	6085.492	3171.793	88.36189	524.4539	297.9399	-359.168	-267.958
50.30072	6090	3543.765	144.9949	782.9167	298.7	-47.2106	-355.758
50.00903	5853.457	1667.425	265.5574	248.6506	152.8273	-863.598	-87.0286
51.03164	6090	3173.788	77.85313	550.4584	298.6954	-337.927	-280.6
50.04948	5853.824	1686.565	263.8036	222.5788	152.5915	-885.152	-81.4136
50.35597	6090	4520.614	145.747	805.911	298.7	172.2457	-391.198
50.29067	6090	3737.456	138.763	778.0172	298.7	-13.4692	-362.055
50.02864	5867.947	1727.868	257.5903	253.5205	159.4132	-852.261	-95.4716
50.14613	6090	3381.141	125.164	734.5489	291.9712	-130.753	-345.159
50.00878	5853.84	1617.657	260.8021	185.7936	154.1669	-936.516	-73.6551
50.24232	6090	3570.258	145.5307	792.5615	298.6774	-33.4439	-360.271
50.42408	6089.943	4671.829	134.1131	823.6926	295.7229	221.8997	-397.913
51.19341	6084.779	3124.795	88.06947	507.7557	294.5391	-385.119	-262.308
51.01316	5774.869	206.1295	251.8298	37.89555	145.8095	-1313.44	54.33086
51.15537	6082.626	3173.675	88.47826	507.5317	297.2963	-375.483	-265.197
50.32454	6089.807	4636.568	163.481	809.6747	298.7	198.3565	-395.233
51.41027	6090	1922.812	102.3146	428.8906	298.7	-702.525	-201.191
50.40496	6089.979	4470.387	147.5135	794.0957	294.5258	151.414	-384.425
51.34177	6088.375	2889.71	111.0097	494.6307	298.7	-444.243	-247.493
51.35466	6090	2591.962	114.5457	453.4114	290.7149	-545.358	-226.92
50.00962	5849.148	1627.925	265.9424	198.1414	152.9792	-920.208	-74.5234
51.32018	6090	2858.555	116.3453	476.3465	281.0313	-469.662	-238.695
51.42712	6090	1910.844	121.7788	451.6991	298.7	-682.098	-203.226
50.14532	6090	3468.696	132.9098	746.9224	290.0005	-100.98	-349.406
50.0279	5867.439	1727.902	257.5881	251.2019	159.4612	-854.377	-94.9172
51.3432	6088.293	3032.474	122.1458	497.9597	298.6985	-412.484	-251.926
50	5842.508	1701.715	273.6757	309.5691	168.7764	-791.84	-101.465
50.94419	6090	3263.145	70.56715	570.3996	298.7	-301.742	-290.798
51.58046	6085.658	2248.706	119.6239	482.804	297.77	-578.493	-214.955
50	5859.327	30	260.9853	244.2827	164.7404	-1198.2	-36.8704
50.49357	6067.079	3625.076	136.0907	622.57	289.8213	-177.592	-313.257
51.1713	6086.583	3106.871	82.05933	490.4228	297.7736	-407.184	-260.722
51.00753	6090	3228.64	75.89364	559.7795	298.7	-318.085	-285.18
50.02459	5853.49	1695.986	271.3198	305.4877	160.9218	-800.982	-100.568
50.13995	6090	3465.326	132.2561	741.3672	290.0427	-107.307	-348.409
50	5846.914	201.216	261.7587	265.3114	165.3245	-1137.78	-43.7712
50.93599	6090	3217.603	67.41796	569.5868	298.7	-311.775	-289.626

51.1983	6086.571	3065.552	84.24924	494.1131	297.7929	-411.249	-259.092
50.39619	6089.853	4489.187	149.6626	801.3108	298.2203	162.1686	-387.533
50	5834.937	230.5947	257.3138	260.0119	183.2809	-1132.44	-45.8551
50.02161	5770.753	30	257.0912	176.1785	166.9412	-1228.94	-1.36908
51.33675	6090	1929.265	123.6805	418.7301	298.7	-713.222	-199.799
50.45646	6090	4719.027	133.8824	825	295.6961	233.2749	-398.803
50.31333	6090	3585.18	144.1757	822.0042	298.7	0.425552	-364.645
51.30874	6090	3008.924	123.4735	492.0592	298.7	-424.517	-251.311
51.29358	6083.621	2685.386	124.405	461.0792	292.2689	-517.759	-231.398
51.31165	6090	2988.622	123.0846	484.2613	298.7	-436.311	-249.02
51.45098	6081.599	2057.726	124.3885	499.3132	296.7923	-601.141	-214.103
50	5858.713	139.5241	263.2693	211.8521	166.849	-1208.54	-34.2535
50.20675	5784.247	30	265.0105	185.7426	178.6893	-1221.62	-3.57402
51.33399	6089.988	2695.38	109.7248	442.1109	289.6798	-536.285	-228.85
50.06939	5871.352	30	263.1036	262.136	163.6738	-1183.99	-40.9047
51.31868	6089.987	2773.417	112.1817	470.6415	289.8761	-492.497	-237.464
50.27769	6090	3547.201	147.1897	785.9666	298.7	-43.9709	-356.99
51.33375	6089.999	1899.327	122.1609	412.9468	298.6854	-725.027	-197.857
50.32853	6090	4642.629	162.0165	800.4324	298.7	190.3502	-393.637
50.00212	5858.572	156.2828	263.0797	231.7483	166.9176	-1185.19	-38.7265
50.00992	5852.538	1624.681	265.5866	248.7508	152.7495	-871.645	-85.3441
51.51996	6086.089	2296.702	107.3327	480.5925	293.6552	-572.221	-217.962
50.02074	5848.13	1681.773	265.0176	232.4345	151.3324	-874.456	-82.2268
Jumlah						30	70

Lampiran 15 Tabel Hasil 4.15 Tabel Hasil 5 Fungsi Objective NSGA-II Kirna dengan 100 Populasi 100 Generasi

Health Pahlawan	Health Musuh	Health Pasukan	Damage Received Pahlawan	Damage Received Musuh	Damage Receive Pasukan	F(1) Agresif	F(2) Supportif	F(3) Leadership	F(4) Egois 1	F(5) Egois 2
50.89171	5419.734	2510.156	4.229786	825	15.80339	86.30886	-80.716	-774.534	-783.709	-1386.84
50.02621	6090	30.20532	285.6	14	112.7919	-1523.57	-31.7985	-1925.24	176.3931	456.1546
50.00028	6090	3355.669	178.4396	431.9578	148.0978	-439.845	-245.362	-516.429	-285.687	-292.682
50.00006	6090	30	285.525	14	117.2518	-1524.21	-33.7716	-1925.94	175.7511	457.1521
53.67716	5483.407	1936.924	3.468158	825	5.325431	1.028631	8.767786	-1000.83	-760.384	-1391.99
65.049	5561.114	55.0181	3.4605	627.0633	67.03358	-379.259	416.3576	-1481.8	-559.774	-987.877
50	6090	30	168.2037	210.762	116.742	-1325.49	-82.7666	-2102.46	-264.512	-229.267
54.05401	5932.669	4800	37.79496	303.22	177.8911	-131.24	-126.479	399.9099	-622.179	-339.777
50.89138	5419.582	2505.405	3.4605	825	13.83417	85.46116	-80.0684	-777.658	-786.24	-1390.47
62.7961	5632.38	31.86253	10.76462	745.1818	12.93434	-339.743	323.1377	-1671.94	-616.567	-1226.77
57.11388	5316.539	1570.78	3.4605	691.5285	21.43911	-67.6776	188.1878	-888.783	-633.774	-1099.32
64.12612	5627.23	151.6966	3.4605	824.2672	26.14851	-208.044	340.3813	-1595.12	-639.448	-1328.58
62.91013	5610.053	30	3.4605	797.4033	12.14239	-276.176	321.3604	-1680.77	-646.677	-1322.02
50.0652	6090	542.7881	206.552	132.7682	144.8097	-1300.73	-86.5938	-1777.66	-116.207	62.26822
56.2655	5929.026	3630.352	53.03655	480.0591	142.7997	-142.252	-44.9754	-161.098	-581.5	-601.103
53.59394	5376.371	92.89192	4.120466	73.089	133.2502	-1078.9	210.8211	-1540.51	-544.721	-308.62
50	6090	1727.237	187.782	294.1445	146.2045	-903.475	-162.243	-1267.93	-220.096	-164.146
58.29345	6089.941	3074.769	65.38285	409.4433	162.3851	-351.006	4.083132	-429.625	-532.502	-482.9
50.72198	5477.958	537.066	225.3058	631.8179	223.0453	-536.285	-32.7993	-1524.03	-15.5952	-426.735
50	6090	688.8974	202.4658	34.68504	129.4566	-1370.57	-70.0499	-1672.76	-106.285	198.0677
56.16974	6033.403	4730.394	27.35202	494.3878	166.781	46.70796	-122.011	291.01	-698.228	-653.292
56.9198	5457.593	30	3.4605	276.4318	27.23526	-853.681	279.9352	-1593.87	-568.174	-616.221
50.13047	6086.911	51.36619	278.544	40.72841	122.0091	-1489.28	-36.9953	-1924.34	148.5272	400.4971

61.13107	6078.372	1919.259	94.77983	824.7246	17.85284	-103.334	88.55649	-1049.32	-492.875	-1083.03
50.00403	6090	89.72297	177.912	45.71983	123.869	-1478.78	-52.7195	-1997.63	-190.829	57.57004
58.05966	6090	3365.778	64.67642	433.2626	154.1631	-273.536	-15.2757	-303.866	-544.333	-509.562
50.0396	6090	1022.142	160.3851	123.5106	156.1595	-1214.04	-108.363	-1572.58	-268.167	-41.7502
50	6086.949	597.5609	207.337	24.04979	127.0993	-1398.24	-63.0808	-1708.31	-86.6264	223.9885
57.18746	6090	2850.693	107.511	411.0557	211.7424	-417.877	-31.6157	-543.779	-409.082	-362.221
61.20357	6078.031	2047.102	102.9963	823.4592	19.83468	-77.609	86.96455	-974.906	-463.998	-1042.9
50.02492	6090	48.99235	173.7438	87.23029	113.8223	-1444.77	-56.7055	-2038.29	-214.788	-24.9058
58.24265	6089.944	3236.13	70.99541	410.9176	160.2486	-318.334	-2.07764	-346.978	-514.924	-457.517
58.85054	6040.588	1493.981	7.156185	485.9761	130.4344	-557.422	74.18854	-1248.56	-725.029	-886.474
50.11395	6090	511.962	196.0262	187.8197	135.7355	-1250.54	-93.5715	-1822.95	-164.325	-60.1049
54.02136	5885.647	4172.17	41.59404	292.7649	164.8237	-248.167	-88.883	122.1598	-595.636	-347.411
50.47729	6090	1764.414	181.9303	299.5598	146.2356	-880.981	-150.742	-1238.49	-233.796	-179.738
55.2399	5324.22	222.1187	24.63182	257.5314	41.27118	-812.77	258.2787	-1445.28	-484.733	-499.132
59.60435	6090	1231.202	3.4605	413.4381	121.2878	-687.814	109.8477	-1359.6	-720.259	-808.44
50	5016.719	363.7382	3.4605	54.21756	158.903	-966.041	180.7663	-1290.52	-506.199	-236.161
50.00317	6089.91	142.7047	265.5466	14	113.3139	-1501.17	-38.0274	-1886.45	109.2253	396.5804
56.04704	5968.967	3947.898	52.1233	498.4201	145.3876	-81.421	-76.5375	-44.9917	-602.397	-623.839
50.02446	6090	30	173.5147	120.9787	114.6357	-1414.83	-63.0719	-2061.41	-223.995	-77.242
50	6090	149.085	256.5734	14	114.0638	-1499.86	-39.3053	-1890.84	79.24485	367.3094
53.87914	5874.316	4342.084	36.91332	294.1889	163.9115	-210.79	-96.4123	204.7759	-610.895	-354.198
50.12899	6090	1761.699	168.6924	361.0901	149.7064	-826.797	-175.436	-1288.2	-298.53	-321.918
50.00246	6073.432	1563.175	180.5803	252.4991	141.9631	-970.789	-143.698	-1328.59	-229.511	-134.126
50.05622	6090	3118.08	176.7964	430.8714	149.8362	-487.332	-236.147	-633.886	-290.053	-310.555
58.61735	6090	1778.515	4.094077	467.5782	149.5008	-544.432	43.69531	-1141.26	-746.487	-859.505
57.07379	6058.629	2528.357	91.92412	461.1825	202.2329	-421.002	-25.2271	-722.287	-467.432	-509.091

57.29511	6083.022	2863.31	99.85807	403.3698	218.0766	-417.958	-27.8523	-531.452	-429.311	-368.93
54.26559	5935.154	4494.781	50.42014	346.3496	178.8053	-146.184	-118.397	246.9379	-588.325	-378.76
56.59207	6090	4383.217	27.27297	513.039	156.0848	-19.0323	-112.803	86.93126	-710.969	-714.659
50.07771	6079.208	1343.025	188.5778	256.6162	138.9532	-1011.69	-134.901	-1434.98	-204.198	-129.463
50	6090	776.3413	217.5567	51.39403	124.6933	-1336.54	-73.7787	-1623.62	-60.1596	227.2313
50.03138	6090	527.9361	203.4224	158.0414	135.7224	-1278.9	-90.0057	-1799.98	-133.465	8.62481
50.00011	6090	48.48007	245.021	30.44896	101.0765	-1503.12	-36.7377	-1958.64	36.62624	292.2282
56.55917	6077.243	4370.349	26.51942	497.2126	165.0746	-32.912	-109.467	94.36997	-706.828	-688.162
50.0339	6090	214.1475	236.7559	14	98.76567	-1485.58	-38.0292	-1875.08	13.69467	300.1469
53.97867	5897.105	4457.165	32.36582	287.2324	162.5996	-202.138	-101.598	249.7176	-628.518	-355.481
50.06123	6082.363	1218.199	190.7305	258.0968	137.7803	-1036.83	-131.827	-1499.03	-198.428	-134.418
59.88401	6090	1194.181	11.07229	411.0188	121.3459	-692.172	120.5743	-1360.31	-690.087	-776.29
59.17177	6090	1186.719	9.297843	383.1062	120.2584	-735.774	105.1805	-1381.17	-699.707	-755.513
58.92557	6090	1079.269	19.34108	374.5651	118.2135	-770.862	104.4667	-1432.54	-667.787	-722.129
58.32652	6089.042	1664.028	3.4605	410.9754	126.3747	-628.953	56.46078	-1188.97	-738.572	-799.173
50.45227	6076.265	2796.526	153.1771	452.9765	142.2806	-515.374	-214.488	-799.935	-364.936	-435.548
50.01957	6090	476.9909	215.6109	188.0706	136.5197	-1259.51	-93.8643	-1827.67	-100.521	0.895793
56.17373	5968.374	3957.363	52.45566	486.8838	145.3612	-88.2883	-70.562	-30.2238	-596.356	-602.124
54.31832	5204.743	1659.528	25.91292	641.7252	14.6396	-109.321	142.9568	-836.759	-560.466	-974.54
54.72783	5159.033	1540.514	26.36955	660.9669	4.46983	-86.4846	169.5383	-858.77	-546.184	-994.265
50.11971	6083.446	1850.852	176.3063	326.5152	153.3634	-841.184	-170.473	-1219.06	-263.007	-235.817
53.52638	5483.318	2063.246	3.4605	825	7.307073	23.2814	-0.47319	-943.069	-762.649	-1385.79
58.4254	6090	1694.589	3.4605	414.9881	126.2504	-617.248	57.39973	-1172.24	-738.331	-801.466
50.46994	6080.483	2806.305	155.9804	385.9736	130.2515	-581.672	-198.496	-769.261	-339.63	-330.56
59.14636	6090	1024.178	15.98922	384.1751	118.3643	-767.801	110.6853	-1458.43	-678.051	-746.914
50	6090	191.3615	173.8162	212.2362	114.6055	-1291.8	-87.4028	-2017.91	-246.172	-202.867

50	6090	144.9518	256.9783	14	115.4738	-1500.72	-39.5099	-1892.43	80.59422	368.9472
58.35101	6089.976	1737.034	3.700223	486.4562	146.8191	-539.115	34.00055	-1180.12	-756.509	-898.294
50	6088.385	923.6944	155.3531	245.3123	115.2091	-1111.29	-119.725	-1679.22	-315.581	-264.558
50.38089	6089.954	1016.37	168.8739	187.8417	118.3827	-1143.53	-100.005	-1585.07	-250.822	-118.609
50.09165	6090	2459.361	164.5471	424.1359	137.6912	-624.696	-209.562	-970.645	-328.669	-389.347
53.75108	5494.249	1854.059	9.609809	789.3144	10.06409	-54.4504	17.42238	-1026.85	-732.592	-1322.85
54.32331	5205.214	1843.15	22.96462	659.59	22.29134	-54.9071	131.0089	-753.913	-574.803	-995.88
50	6090	72.88624	235.3087	14	102.307	-1514.55	-35.4024	-1947.84	8.362308	286.6442
50.38442	6089.953	961.0553	168.9711	179.512	117.2422	-1162.84	-96.0771	-1609.3	-248.363	-109.864
55.47533	5325.593	222.0197	24.06705	270.382	41.69963	-795.801	262.2704	-1442.99	-486.64	-515.76
54.25919	5175.044	1668.169	19.02582	653.5751	13.99999	-84.4263	145.5461	-825.401	-579.848	-1008.71
59.39385	6090	1040.985	19.79799	404.8787	119.8135	-738.874	113.3398	-1445.71	-666.819	-758.623
58.32405	6090	1736.442	3.4605	512.1066	146.6895	-514.127	28.08996	-1191.92	-764.131	-938.205
58.17952	6089.68	3133.864	74.81089	408.0066	159.354	-342.9	0.641994	-396.048	-502.359	-448.804
58.45066	6090	1823.638	3.531771	527.779	146.227	-478.476	25.97643	-1149.83	-765.912	-953.316
56.40903	6064.859	2519.691	93.14864	483.0225	197.9974	-416.736	-49.575	-763.841	-480.339	-554.894
57.46101	6063.497	2318.922	80.42058	464.2423	202.9101	-453.932	-9.59742	-826.471	-501.951	-559.19
50.13285	6090	30	165.0256	177.2787	110.6741	-1356.16	-70.7312	-2087.34	-264.742	-189.406
50	6090	516.7783	203.638	186.0062	136.2133	-1253.8	-96.2809	-1817.69	-140.208	-33.7788
58.1623	6089.514	3283.219	73.18656	401.9864	160.2507	-319.312	-3.98213	-320.761	-506.486	-435.175
50	6090	1009.662	150.7107	249.722	115.1764	-1090.28	-124.254	-1642.95	-332.562	-281.333
50.04789	6090	30	239.221	34.10675	100.2583	-1502.09	-35.6851	-1972.47	17.09494	266.8043
50	6090	772.3862	216.8553	50.25275	124.4935	-1338.46	-73.4238	-1625.75	-62.2122	226.2617
50.03313	6089.114	597.727	211.0929	145.7652	144.5982	-1277.09	-91.3304	-1752.24	-104.58	61.06692
50.47534	6080.318	2889.761	157.3722	409.5923	130.3909	-541.211	-205.72	-735.494	-340.774	-355.58
Jumlah						3	65	9	1	22

BIOGRAFI PENULIS



Cindy Taurusta, Putri Pertama dari 4 Bersaudara ini lahir di Surabaya, 25 April 1990. Mengenyam pendidikan dasar di SDN Margorejo I Surabaya (Kelas 1 – 4) dan SDN Kebraon II Surabaya (Kelas 5 - 6), lalu melanjutkan ke SMPN 16 Surabaya dan SMAN 9 Surabaya. Tahun 2008 menempuh pendidikan D3 Teknik Informatika Politeknik Elektronika Negeri Surabaya – ITS dan lulus tahun 2011. Setelah menyelesaikan tahap D3 meneruskan tahap D4 di tempat yang sama yaitu Teknik Informatika Politeknik Elektronika Negeri Surabaya yang lulus pada tahun 2013.

Untuk mencapai cita - cita penulis untuk menjadi pengajar di lingkup Perguruan Tinggi, penulis meneruskan pendidikan Magister pada tahun 2013 di Institute Teknologi Sepuluh Nopember Surabaya dengan mengambil bidang studi Jaringan Cerdas Multimedia dan mempelajari bidang keahlian Teknologi Permainan dan menyelesaikannya pada tahun 2015.

Contact person :

Email : taurustacindy@gmail.com

“I can do the Impossible, I can Reach the Unreachable because ALLAH always beside Me”